# REPORT DOCUMENTATION PAGE

Form Approved
OMB No. 0704-0188



| 1. REPORT DATE *(DD-MM-YYYY)* | 2. REPORT TYPE | 3. DATES COVERED *(From - To)* |
|---|---|---|
| | Technical Papers | |

**4. TITLE AND SUBTITLE**

Please see attached

**5a. CONTRACT NUMBER** N/A

**5b. GRANT NUMBER**

**5c. PROGRAM ELEMENT NUMBER** 61102F

**6. AUTHOR(S)**

**5d. PROJECT NUMBER** 2308

**5e. TASK NUMBER** m19B

**5f. WORK UNIT NUMBER** 346058

**7. PERFORMING ORGANIZATION NAME(S) AND ADDRESS(ES)**

Air Force Research Laboratory (AFMC)
AFRL/PRS
5 Pollux Drive
Edwards AFB CA 93524-7048

**8. PERFORMING ORGANIZATION REPORT**

**9. SPONSORING / MONITORING AGENCY NAME(S) AND ADDRESS(ES)**

Air Force Research Laboratory (AFMC)
AFRL/PRS
5 Pollux Drive
Edwards AFB CA 93524-7048

**10. SPONSOR/MONITOR'S ACRONYM(S)**

**11. SPONSOR/MONITOR'S NUMBER(S)** AFRL-PR-ED-TP-2000-161

**12. DISTRIBUTION / AVAILABILITY STATEMENT**

Approved for public release; distribution unlimited.

**13. SUPPLEMENTARY NOTES**

**14. ABSTRACT**



**15. SUBJECT TERMS**

| 16. SECURITY CLASSIFICATION OF: | | | 17. LIMITATION OF ABSTRACT | 18. NUMBER OF PAGES | 19a. NAME OF RESPONSIBLE PERSON |
|---|---|---|---|---|---|
| | | | | | Leilani Richardson |
| a. REPORT | b. ABSTRACT | c. THIS PAGE | A | | 19b. TELEPHONE NUMBER *(include area code)* |
| Unclassified | Unclassified | Unclassified | | | (661) 275-5015 |

2308m19B

MEMORANDUM FOR PRS (In-House/Contractor Publication)

FROM: PROI (TI) (STINFO) 07 Aug 2000

SUBJECT: Authorization for Release of Technical Information, Control Number: **AFRL-PR-ED-TP-2000-161**
D. C. Wadsworth, D. B. VanGilder (ERC); I. J. Wysong (AFRL/PRSA); C. Kaplan, D. Mott (NRL/LCPFD), "SUPREM-DSMC Version 1.0 User's Manual (Draft)"

**HPCMO CHSSI Software Acceptance Test Review** **(Statement A)**
**(Washington DC, 11 Aug 00) (Submission Deadline: 10 Aug 00)**

# *DRAFT*[1]
# SUPREM-DSMC Version 1.0
# User's Manual[2]

D. C. Wadsworth, D. B. VanGilder, ERC Inc.
I. J. Wysong AFRL/PRSA
C. Kaplan, D. Mott NRL/LCPFD

July 19, 2000

[1]"Approved for public release; distribution unlimited"

# Abstract

SUPREM-DSMC is a scalable, parallel, reacting-flow Direct Simulation Monte Carlo code for modeling nonequilibrium and collisional gas dynamic processes in complex geometries of interest to DoD. This manual is intended to document installation and operation of the SUPREM-DSMC package. This manual is presently under development. For the most part, only the required user-generated input files for the package are documented herein. Not all of the input variables documented at present will be available in all versions of the software.

# Contents


**1 Introduction** 11
1.1 Quick-start guide . . . . . 11

**2 Obtaining and Installing SUPREM-DSMC** 13
2.1 Hardware and Software Requirements . . . . . 13
2.2 Obtaining the Software . . . . . 13
2.3 Installing the Software . . . . . 13
2.4 Uninstalling the Software . . . . . 13

**3 Input Files** 15
3.1 Notation . . . . . 15
3.2 Execution Control File . . . . . 15
3.2.1 Execution Control File Template . . . . . 16
3.2.2 Keyword Definition . . . . . 18
3.2.2.1 User Level and Problem Description . . . . . 18
3.2.2.2 Solution Quality Control . . . . . 18
3.2.2.3 Job Control . . . . . 24
3.2.2.4 Run Type and Input File Definition . . . . . 26
3.2.2.5 Reference and Initial Condition Definition . . . . 27
3.2.2.6 Grid Definition . . . . . 32
3.2.2.7 Time Step Definition . . . . . 36
3.2.2.8 Algorithm Control . . . . . 40
3.2.2.9 Output Control . . . . . 47
3.3 Geometry File . . . . . 49
3.3.1 Geometry File Template . . . . . 50
3.3.2 Keyword Definition . . . . . 50
3.4 Gas File . . . . . 50
3.4.1 Gas File Template . . . . . 50
3.4.2 Keyword Definition . . . . . 51
3.4.2.1 Species Definition . . . . . 51
3.4.2.2 Bimolecular Collision Event Definition . . . . . 54
3.4.2.3 Trimolecular Collision Event Definition . . . . . 59
3.4.2.4 Unimolecular Event Definition . . . . . 60
3.5 Boundary Condition File . . . . . 60

3.5.1 Boundary Condition File Template . . . . . 61
3.5.2 Keyword Definition . . . . . 61

**4 Code Operation** **75**
4.1 Pre-processing with Pre-SUPREM . . . . . 75
4.2 SUPREM . . . . . 75
4.3 Post-processing with Post-SUPREM . . . . . 75

**5 Demonstration Problems** **77**
5.1 Reentry Vehicle Flowfield . . . . . 77
5.2 Microvalve Flow . . . . . 77

**6 Frequently Asked Questions, Tips and Troubleshooting** **79**

**7 Utilities** **81**

**8 Acknowledgement** **83**

**9 Bibliography** **85**

**A Sample Input Files** **87**
A.1 Execution Control (.in) Files . . . . . 87
A.2 Geometry (.geo) Files . . . . . 87
A.3 Gas (.gas) Files . . . . . 87
A.4 Boundary Condition (.bc) Files . . . . . 87

**B Sample Output Files** **89**
B.1 Pre-Processor Output . . . . . 89
B.2 SUPREM Output . . . . . 89
B.3 Post-Processor Output . . . . . 89

**C Error Messages** **91**

# List of Figures

4.1 Flowchart for operation of the SUPREM package. (a) pre-processing with Pre-SUPREM, (b) running SUPREM, (c) post-processing with Post-SUPREM, (d) restarting an existing SUPREM run, (e) refining an existing SUPREM run via Pre-SUPREM. . . . . 75

~~LIST OF FIGURES~~ ?

Intentionally left blank

# List of Tables

2.1 Supported Architectures . . . . . 13

LIST OF TABLES

Intentionally left blank.

# Chapter 1

# Introduction

## 1.1 Quick-start guide

Intentionally left blank

# Chapter 2

# Obtaining and Installing SUPREM-DSMC

## 2.1 Hardware and Software Requirements

The SUPREM package is designed for operation on a variety of architectures available throughout the HPC. The package is coded in fortran90 and requires the MPI[] library. Table 2.1summarizes the systems which are presently supported. Chapter 6 discusses in more detail any known issues regarding implementation on specific systems.

## 2.2 Obtaining the Software

## 2.3 Installing the Software

## 2.4 Uninstalling the Software

| Architecture | Operating System | Compiler | Comments |
|---|---|---|---|
| IBM SP | | | |
| SGI Origin | | | |
| PentiumIII | Linux 2.2.5-15smp | pgf90 3.1-2 | |
| Compaq Alpha | | | |

Table 2.1: Supported Architectures

Intentionally left blank

# Chapter 3

# Input Files

A user must create four major input files to perform a simulation with the SUPREM package: the execution control file, the geometry file, the gas file, and the boundary condition file. Each file is organized hierarchically and is keyword-based. The files are purposely quite detailed, and allow a knowledgeable user to control the simulation to a very detailed level. For essentially all major features, however, the code can estimate proper values automatically. In either case, the code is not intended to be a 'black box' - the details of the numerical algorithm, physics, accuracy, and other key features a user needs to know to compactly describe and archive a given run of the code are directly available, and self-documenting, in the input (and output) files of the run.

At present, the input files must be created using a text editor, however, the structure naturally lends itself to the graphical user interface (GUI) to be implemented later in the project.

## 3.1 Notation

In the following, **BOLD** indicates a reserved keyword, **bold** indicates a reserved keyword dependent on previous keywords in the definition, *italic* text indicates a user-supplied variable, *int* denotes an integer, $int_+$ a positive integer, *real* denotes a real number, $real_+$ a positive real, and *string* (or *string1*, ... *stringn*) indicates a blank-delimited character string. Angle brackets <...> indicate optional text. The notation [a | b | c] denotes that a single value a, b, or c must be selected. Indenting is utilized for clarity, but need not be present. Keywords must be input in capital letters only.

## 3.2 Execution Control File

The execution control file defines basic numerical and job-related parameters required for a specific run of the code. The file is divided into several independent sections.

### 3.2.1 Execution Control File Template

```
#### SUPREM Version 1.0 Exec Control File ###
user-supplied file description string(s)
USER_LEVEL [ NOVICE | INTERMEDIATE | EXPERT]
PROBLEM_TYPE [ string | COMPRESSION | EXPANSION | STP| SUPERSONIC_BODY | CHANNEL_FLOW]
SOLN_QUALITY_CONTROL [AUTO | MANUAL]
  QUALITY [TIME_STEP | COLLN_SEPARATION | CELL_SIZE | BDRY_ELEMENT_SIZE |
           CELL_SAMPLE_SIZE | BDRY_SAMPLE_SIZE | PARALLEL]
    METRIC name
    GLOBAL
      MULTIPLIER real+
      DESIRED_PCT real+
      ABORT_PCT real+
    LOCAL
      WORST_MULTIPLIER real+
      ABORT_PCT real+
 <QUALITY repeats as necessary for remaining types>
REALISM [GAS | COLLN | BOUNDARY]
   METRIC name
   GLOBAL
     MULTIPLIER real+
     DESIRED_PCT real+
     ABORT_PCT real+
   LOCAL
     WORST_MULTIPLIER real+
     ABORT_PCT real+
<REALISM repeats as necessary for remaining types)>
JOB_CONTROL
  MAX_WALL_CLOCK_HRS real
  MAX_MEMORY_MB real+
  MAX_DISK_MB real+
  NUM_PROCESSES int+
  <HOST_LIST string1 stringn>
  PARTITIONING
    INITIAL [AUTO | MANUAL | LIST]
      <type-dependent parameters>
    ADAPTION [OFF | MANUAL | AUTO]
      <type-dependent parameters>
    DIAGNOSTICS [OFF | MIN | MORE | MAX]
RUN_TYPE [NEW | RESTART | REFINE]
  <type-dependent parameters>
SUPREM_UNITS [SI]
REF_CONDS
  GAS [AMBIENT | BC_SOURCE]
    <type-dependent parameters>
  SCALING
    <optional parameters>
INITIAL_CONDS [VACUUM | GAS_REF_CONDS| MANUAL]
  <type-dependent parameters>
```

```
RANDOM_SEED [int+ | AUTO]
GRID_CONTROL
  LVL1_SPACING [AUTO | MANUAL | LIST | REFINE]
    <type-dependent parameters>
  LVL2_SPACING
    INITIAL [AUTO | LIST | REFINE]
     <type-dependent parameters>
    ADAPTION [OFF | AUTO | MANUAL]
     <type-dependent parameters>
  LVL2_WEIGHT_FACTORS
    INITIAL [AUTO | MANUAL | LIST | REFINE]
      <type-dependent parameters>
    ADAPTION [OFF | AUTO | MANUAL]
      <type-dependent parameters>
  DIAGNOSTICS [OFF | MIN | MORE | MAX]
TIME_DEP [STEADY | UNSTEADY]
  TIME_STEP_CONTROL
    INITIAL [AUTO | MANUAL | LIST | REFINE]
      <type-dependent parameters>
    ADAPTION [OFF | AUTO | MANUAL]
      <type-dependent parameters>
  UNSTEADY_PHASE
    INIT_TIME real
    RESET_SAMPLE [AUTO | MANUAL]
    COMPLETION_CRITERIA [AUTO | MANUAL]
      <type-dependent parameters>
  STEADY_PHASE
    COMPLETION_CRITERIA [AUTO | MANUAL]
      <type-dependent parameters>
  DIAGNOSTICS [OFF | MIN | MORE | MAX]
ALGORITHM_CONTROL
  SPECIES_WEIGHT_FACTORS [OFF | AUTO | MANUAL]
  <BACKGROUND_SPECIES string>
  COLLISIONS [ON | OFF]
    WEIGHT_FACTORS [OFF | AUTO | MANUAL]
    DIAGNOSTICS [OFF | MIN | MORE | MAX]
  MOVEMENT [ON | OFF]
    EXTERNAL_FORCE [OFF | GRAVITY]
    DIAGNOSTICS [OFF | MIN | MORE | MAX]
  SAMPLE_CELL [ON | OFF]
    <[XYZ_LIST string | XYZ_BOX string | LVL1_LIST string]>
      <list descriptors>
      <TYPE [DETAIL | MACRO | INTEGRATED | MOL_LIST]>
       <type-dependent parameters>
    <location list and type repeat as necessary>
  SAMPLE_COLLN [ON | OFF]
    <[XYZ_LIST string | XYZ_BOX string | LVL1_LIST string]>
      <list descriptors>
      <TYPE [DETAIL | MACRO | MOL_LIST]>
        <type-dependent parameters>
```

```
    <location list and type repeat as necessary>
  SAMPLE_BDRY [ON | OFF]
    <[XYZ_LIST string | ENTITY_LIST string]>
      <list descriptors>
      <TYPE [DETAIL | MACRO | INTEGRATED | MOL_LIST]>
       <type-dependent parameters>
    <location list and type repeat as necessary>
  OUTPUT [ON | OFF]
   PREFIX string
   WRITE [STDOUT | RESTART | SAMPLE_LIST] [OFF |AUTO | MANUAL]
   <type-dependent parameters>
#### SUPREM Version 1.0 Exec Control File end ###
```

### 3.2.2 Keyword Definition

#### 3.2.2.1 User Level and Problem Description

This short section serves to restrict the keywords which can appear in the input files, and to initialize default values.

**USER_LEVEL [ NOVICE | INTERMEDIATE | EXPERT]**

SUPREM-DSMC contains several features which, if utilized, can allow the user improved control over the simulation, but at the possible cost of decreased solution accuracy, realism, or efficiency. The user level serves as a filter for features as they may be accessed through the input files, and as a filter for code automation, error-checking, and diagnostics. An EXPERT user is allowed access to essentially all input parameters, and is allowed to supply relatively arbitrary input values, or over-ride any recommended inputs. A NOVICE user is assumed to be one with little knowledge of DSMC methods, for which only more robust models and suitably reasonable input values are made available. An INTERMEDIATE user has a wider range of input parameters to select from than the NOVICE, but is not allowed to over-ride recommended values.

**PROBLEM_TYPE [ string | COMPRESSION | EXPANSION | STP | SUPERSONIC_BODY | CHANNEL_FLOW]**

This keyword allows the user to classify the basic type of problem, and thus guide the code (or GUI) in defining the problem setup. Problems can be broadly categorized as COMPRESSION, involving the formation of a shock layer around a body, EXPANSION, involving the expansion of a high pressure gas, and STP, involving a gas at standard temperature and pressure (e.g., 'room' conditions). The two remaining specific pre-defined types are representative of the demonstration problems. Other types may be made available in the future.

#### 3.2.2.2 Solution Quality Control

This section allows the user to define the requirements the solution should or must meet, and also serves to establish criteria to guide the adaption features.

These requirements can be grouped into DSMC-specific numerical requirements, computational efficiency requirements, and physical fidelity requirements.

A high quality DSMC numerical solution would possess:

1. suitably small timestep
2. suitably small collision selection distance (nearest-neighbor collisions)
3. sufficient number and size of sampling cells and boundary elements to adequately resolve the macroscopic properties and control or limit statistical noise.

An efficient parallel simulation would possess:

1. minimum necessary number of cells, molecules and timesteps
2. good parallel efficiency or load-balancing

An accurate, valid simulation of the physics in an arbitrarily complex flowfield would satisfy:

1. appropriate representation of the gas species in the flowfield (e.g., number and type of modes or states available)
2. appropriate representation of gas-gas collision processes
3. appropriate representation of boundary conditions (including gas-boundary collision processes, or mechanisms for accurate introduction or removal of molecules from the simulation)

Each of these three major requirements is treated similarly - each is quantified by a suitable local metric in the simulation. A criteria is evaluated based on the ratio of the metric and the local simulation property, with the criteria scaled to be optimal at unity, poor when larger then unity, and overly conservative when less than unity. The user defines the requirements the simulation must satisfy globally (over the complete domain) and locally (worst-case) in terms of the distribution of this metric or criteria. The global quality measure is defined by the desired percentage of the domain which must satisfy the metric (or some multiple of the metric), and the percentage that, if exceeded, will cause the run to abort. The local quality measure is defined by the worst case value (multiple) of the metric allowed anywhere in the domain, and the percentage of the domain that will be allowed to exceed this value before causing an abort of the run.

**SOLN_QUALITY_CONTROL** [ AUTO | MANUAL ]

This is the header line for the quality and realism subsections. Each subsection has a relatively conservative set of default, or AUTOmatic, values. The MANUAL option allows the EXPERT user to over-ride the default values for one or more subsections.

**QUALITY** [TIME_STEP | COLLN_SEPARATION | CELL_SIZE | BDRY_ELEMENT_SIZE | CELL_SAMPLE_SIZE | BDRY_SAMPLE_SIZE | PARALLEL]

The user can define the requirements that must be satisfied by the time step, the collision separation distance achieved, the size of the grid cells and boundary elements, the cumulative sample size achieved in the grid cells and boundary elements, and the parallel efficiency of the run. For each QUALITY type, the user can define the global and local requirements based on a type-dependent metric.

**QUALITY** type

**METRIC** name

Each QUALITY type has one or more metrics or derived parameters that are used to quantify the type. The metric will be compared with the actual parameters used in the simulation to evaluate the quality based on a pass-fail criteria.

**GLOBAL**

This is the header for the description of the global or overall features the simulation must satisfy in terms of the selected metric.

**MULTIPLIER** $real_+$

This is a pre-multiplier to be applied to the metric. The metrics are nominally defined such that unity is a desirable value of the quality criteria. The multiplier allows this to be adjusted as necessary.

**DESIRED_PCT** $real_+$

This is the global percentage of the simulation that should meet the quality criteria. This is a 'soft' limit used to guide the code adaption features. For a cell-based type, each sampling, or 'level 2', cell contributes equally to the global value. For a boundary element type, each boundary element, as discretized in the level 2 grid, contributes equally. For the parallel type, each processor contributes equally.

**ABORT_PCT** $real_+$

This is the maximum global percentage of the simulation that is allowed to fail the (MULTIPLIER × metric) criteria before the simulation will be aborted. Conservatively, ABORT_PCT = 100 - DESIRED_PCT, but this need not be the case.

**LOCAL**

This is the header for the description of the local or worst-case values of the metric that are allowable.

**WORST_MULTIPLIER** *real*$_+$

This is a pre-multiplier to be applied to the metric to define the limiting or worst-case value that can be tolerated in the simulation. To be useful, the worst multiplier must be greater than the global multiplier.

**ABORT_PCT** *real*$_+$

This is the maximum (global) percentage of the simulation that is allowed to fail the worst-case criteria before the simulation will be aborted. To be useful, the local abort percentage value must be less than the global abort percentage.

**QUALITY** TIME_STEP

**METRIC** [COLLN_FREQ]

The metric is the mass-averaged collision frequency in the cell. Others may be added in the future. The default values (shown below) request that 100 percent of the cells in the simulation use a timestep that is less than or equal to one-half of the local collision frequency. The simulation will abort if one percent of the simulation fails this test, or if 0.5 percent of the cells use a timestep that is greater than the collision frequency.

**GLOBAL**

**MULTIPLIER** 0.5
**DESIRED_PCT** 100.0
**ABORT_PCT** 1.0

**LOCAL**

**WORST_MULTIPLIER** 1.0
**ABORT_PCT** 0.5

**QUALITY** COLLN_SEPARATION

**METRIC** [MEAN_FREE_PATH]

The metric is the mass-averaged mean free path in the cell. Others may be added in the future. The default values request that 100 percent of the cells in the simulation achieve a collision separation distance that is less than or equal to the one-half of a mean free path. The simulation will abort if one percent of the simulation fails this test, or if 0.5 percent of the cells achieve a separation distance that is greater than the mean free path.

**GLOBAL**

**MULTIPLIER** 0.5
**DESIRED_PCT** 100.0
**ABORT_PCT** 1.0

**LOCAL**

**WORST_MULTIPLIER** 1.0
**ABORT_PCT** 0.5

**QUALITY** CELL_SIZE

**METRIC** [MEAN_FREE_PATH]

The metric is the mass-averaged mean free path in the cell. Others may be added in the future. The default values (shown below) request that 100 percent of the cells in the simulation have a maximum dimension that is less than or equal to one mean free path. The simulation will abort if one percent of the simulation fails this test, or if 0.5 percent of the cells are larger than two mean free paths.

**GLOBAL**

**MULTIPLIER** 1.0
**DESIRED_PCT** 100.0
**ABORT_PCT** 1.0

**LOCAL**

**WORST_MULTIPLIER** 2.0
**ABORT_PCT** 0.5

**QUALITY** BDRY_ELEMENT_SIZE

**METRIC** [MEAN_FREE_PATH]

The metric is the mass-averaged mean free path in the cell containing the element. Others may be added in the future. The default values (shown below) request that 100 percent of the elements in the simulation have a maximum dimension that is less than or equal to one mean free path. The simulation will abort if one percent of the simulation fails this test, or if 0.5 percent of the elements are larger than two mean free paths.

**GLOBAL**

**MULTIPLIER** 1.0
**DESIRED_PCT** 100.0
**ABORT_PCT** 1.0

**LOCAL**

**WORST_MULTIPLIER** 2.0
**ABORT_PCT** 0.5

**QUALITY** CELL_SAMPLE_SIZE

**METRIC** [ONE_PCT_SCATTER]

The metric is the sample size (the total number of molecules sampled) in a cell required to achieve a standard deviation of the statistical fluctuation of one percent, assuming a normal distribution of cell population. Others may be added in the future. The default values (shown below) request that 100 percent of the cells in the simulation have a sample size such that the standard deviation is 0.1 percent. The abort parameters and the worst multiplier are not particularly useful in controlling the simulation at present, but are expected to be useful in the future for estimating run times.

**GLOBAL**

**MULTIPLIER** 0.1
**DESIRED_PCT** 100.0
**ABORT_PCT** 100.0

**LOCAL**

**WORST_MULTIPLIER** 2.0
**ABORT_PCT** 100.0

**QUALITY** BDRY_SAMPLE_SIZE

**METRIC** [ONE_PCT_SCATTER]

The metric is the sample size (total number of incident molecules) at a boundary element required to achieve a standard deviation of the statistical fluctuation of one percent, assuming a normal distribution of flux. Others may be added in the future. The default values (shown below) request that 100 percent of the elements in the simulation have a sample size such that the standard deviation is 0.1 percent. The abort parameters and the worst multiplier are not particularly useful in controlling the simulation at present, but are expected to be useful in the future for estimating run times.

**GLOBAL**

**MULTIPLIER** 0.1
**DESIRED_PCT** 100.0
**ABORT_PCT** 100.0

**LOCAL**

**WORST_MULTIPLIER** 2.0
**ABORT_PCT** 100.0

**QUALITY** PARALLEL

**METRIC** [CPU_100PCT_ACTIVE]

The metric is the percentage of active processing time on each processor; others may be added in the future. For this metric, the multiplier parameters give the fraction of active time ($0 <$ MULTIPLIER $\leq 1.0$, and thus have a slightly different meaning than for the other QUALITY types. The default values (shown below) request that 100 percent of the processors used in the run are active 90 percent of the time (10 percent net idle time). The simulation will abort if 20 percent of the processors fail this test, or if 10 percent of the processors are active 50 percent of the time.

**GLOBAL**
**MULTIPLIER** 0.9
**DESIRED_PCT** 100.0
**ABORT_PCT** 20.0

**LOCAL**
**WORST_MULTIPLIER** 0.5
**ABORT_PCT** 10.0

#### 3.2.2.3 Job Control

This section defines parameters that define the computational requirements of the job. Many of these parameters may be present in the specific script file used to submit the job onto a batch system, but they will also be needed by, e.g., the adaption features of the code.

**JOB_CONTROL**

This is the section header.

**MAX_WALL_CLOCK_HRS** *real*

This is maximum number of wall clock hours allowed for the job. A negative value means there is no limit; a value of zero causes the code to read the data files, initialize the problem, write the data files, and then exit.

**MAX_MEMORY_MB** *real*

This is maximum number of megabytes of memory allowed for the job. A non-positive value means there is no limit.

**NUM_PROCESSES** $int_+$

This is number of processes or nodes that will be used for the job. A specific list of hostnames on which to run the job can be supplied.

**<HOST_LIST** *string1 ... stringn*>

This is an optional line which lists specific hostnames on which to run the job. A process is started on each hostname, in left-to-right order. If the number of hostnames is less than NUM_PROCESSES, the list is repeatedly traversed as necessary.

**PARTITIONING**

This is the header for the section which defines the partitioning of the computational domain (groups of level 1 cells) to processes (domain decomposition).

**INITIAL** [AUTO | MANUAL | LIST]

This section establishes the initial partitioning of the domain.

**INITIAL** AUTO

The AUTO option assumes the computational effort is uniform over the domain and decomposes the domain into several equal volume partitions per process.

**INITIAL** MANUAL

The MANUAL option allows the EXPERT user to prescribe the number of partitions (parallel granularity), their shape (clustering of cells), and their distribution over processors subject to the available models.

**NUMBER** $int_+$

This is the number of partitions, $1 \leq$ NUM_PROCESSES $\leq$ NUMBER.

**SHAPE** [CONSEC_CELLS | CONST_VOL | CONST_MOL]

This describes the shape of a partition.

**CONSEC_CELLS**

A partition consists of sequence (a stride of unity in the cell array) of level 1 cells.

**CONST_VOL**

A partition consists of a cluster of level 1 cells such that the variation across partitions of enclosed volume is minimized, while the volumes are as cubical as possible.

**CONST_MOL**

A partition consists of a cluster of level 1 cells such that the variation across partitions of the number of simulated molecules contained within the volume is minimized, while the volumes are as cubical as possible.

**DISTRIBUTE** [CYCLIC | BLOCK | RANDOM]

This determines how partitions are assigned to processes. CYCLIC assigns partitions sequentially, one per process, repeating the traverse as necessary until all partitions have been assigned processes. BLOCK assigns a group of consecutive partitions of approximate size NUMBER / NUM_PROCESSES to each processor. RANDOM assigns partitions to processes randomly, subject to the requirement that the number of partitions per process is as uniform as possible.

**INITIAL LIST**

The LIST option allows the EXPERT user to provide a list assigning level 1 cells to processors, one assignment per line.

**LVL1_CELL** $int_+$ **PROCESS** $int_+$

**ADAPTION** [OFF | AUTO | MANUAL ]

This section prescribes the methods for dynamic adaption (load-balancing), including redistribution of partitions to processors or redefinition of partitions.

**ADAPTION** OFF

The initial partition is maintained during the simulation.

**ADAPTION** AUTO

Partitions are maintained as originally defined, but redistributed among processors to attempt to improve load-balance as necessary to satisfy solution quality requirements (see also QUALITY PARALLEL).

**ADAPTION** MANUAL

The user defines the frequency at which the load-balance is evaluated, and the model used to redefine and redistribute partitions. (See also QUALITY PARALLEL).

**DIAGNOSTICS** [OFF | MIN | MORE | MAX]

This controls the level of run-time diagnostics that are output to stdout during partitioning.

#### 3.2.2.4 Run Type and Input File Definition

**RUN_TYPE** [NEW | RESTART | REFINE]

This determines the type of input files that are to be expected by the codes. These files must reside in the current directory.

**RUN_TYPE** NEW

The user must supply filenames of the geometry, gas, and boundary condition files.

**GEOM_FILE** *string*

This is the name of the geometry file (see 3.3)

**GAS_FILE** *string*

This is the name of the gas file (see 3.4)

**BC_FILE** *string*

This is the name of the boundary condition file (see 3.5)

**RUN_TYPE** RESTART

The code will read existing data files from previous run. The user must supply the filename prefix used to generate the files. The code will attempt to identify whether the files were written in formatted or unformatted form.

**PREFIX** *string*

This is the prefix used to name the files in the previous run (see OUTPUT PREFIX, below)

**RUN_TYPE** REFINE

This indicates that one or more data files generated by a previous run are to be read and potentially modified or used as reference data to allow improvement of the simulation. The user must supply the prefix used to name the files.

**PREFIX** *string*

#### 3.2.2.5 Reference and Initial Condition Definition

This section various reference quantities and prescribes the initialization of numerical parameters, and the initialization of molecules in the flowfield.

**SUPREM_UNITS** [SI]

This defines the units in which all dimensional quantities are input and output. Only the SI (or MKS, meter-kilogram-second) system is presently available.

**REF_CONDS**

This is the header for the definition of gas reference properties and geometrical reference properties.

**GAS** [AMBIENT | BC_SOURCE]

This selects the method used to define the gas reference properties.

**GAS AMBIENT**

In this case, the gas is defined in terms of 'bulk' (or mass-averaged, in the case of multi-species) macroscopic properties. The user must supply a minimal set of properties, selected from several common variables.

**[NUMBER_DENSITY** *real*$_+$ | **MASS_DENSITY** *real*$_+$ | **MOLAR_DENSITY** *real*$_+$ | **PRESSURE** *real*$_+$]

The user must supply either the number density $n$, mass density $\rho$, or molar density $c$, or the (static) pressure $p$, of the gas. These quantities are defined as $\rho = mn$, $p = nk_BT$, $c = \rho/M$, where $m$ is the (mean) molecular mass, $M$ is the (mean) molar molecular weight, $k_B$ is the Boltzmann constant.

**TEMP** *real*$_+$

This is the reference temperature, $T_{\text{ref}}$.

**VEL_X** *real* **VEL_Y** *real* **VEL_Z** *real*

This is the reference velocity vector, $V_{\text{ref}}$, with one component for each geometrical dimension of the problem. If the total magnitude is zero, then any reference quantity requiring a non-zero reference speed will utilize the acoustic sound speed in the gas, $a_{\text{ref}} = \sqrt{2RT_{\text{ref}}}$, where $R$ is the gas constant.

For a multi-species problem, the user must also supply a mean property and the relative concentration of one or more species.

**<[MEAN_MOLEC_WT** *real*$_+$ | **GAS_CONSTANT** *real*$_+$]>

The user must supply either the mean molecular weight of the gas, $M$, or the representative gas constant $R = k_B/m$.

**<SPECIES** *string* [MASS_FRACTION *real*$_+$ | MOLE_FRACTION *real*$_+$ | PARTIAL_PRESSURE *real*$_+$]>

This line defines the reference value of the relative population of a species in a multiple species problem. This line is repeated for each species for which a non-zero reference population is desired. At least one such line must occur for a multi-species problem. The quantities for a species $i$ are defined as: mass fraction $\omega_i = \rho_i/\rho$ mole fraction $x_i = c_i/c$, partial pressure $p_i = n_i k_B T$, where $n_i$ and $\rho_i$ are the number density and mass density of the species.

**GAS BC_SOURCE**

This denotes that the reference conditions will be set equal to the ambient values defined at the named SOURCE boundary condition in the boundary

condition file (Section 3.5). The user must identify the source in terms of the named geometrical ENTITY (or COMPONENT) to which it is applied.

**ENTITY** *string*
**<COMPONENT** *string>*

**SCALING**

This is the header line for a subsection in which parameters useful for scaling or normalizing solution quantities may be defined. Nominal geometrical lengths, areas, and other parameters useful for scaling can be determined from the geometry file 3.3, and the gas reference conditions. The user may optionally provide one or more values to over-ride the nominal defaults.

**REF_LENGTH** $real_+$

The default reference length is the maximum extent of any single geometrical entity in any coordinate direction.

**REF_AREA** $real_+$

The default reference area is the maximum cross sectional area of any single geometrical entity in a plane normal to the x coordinate axis.

**MOMENT_REF_PT** *real real real*

The default moment reference point is the coordinate origin.

**MOMENT_REF_LENGTH** $real_+$ $real_+$ $real_+$

The default moment reference length for each coordinate axis is the maximum extent of any geometrical entity along that axis.

**MIN_LENGTH** *real*

The minimum length parameter is used to constrain the minimum cell and boundary element dimension. The default value is REF_LENGTH / 200. For the case of discretized geometries, the value is 20 percent of the minimum node spacing (the minimum separation of one node from another). A non-positive value means the minimum dimension is unconstrained.

**MIN_AREA** *real*

The minimum area parameter is used to constrain the minimum boundary element area. The default value is (MIN_LENGTH)$^2$. A non-positive value means the minimum element area is unconstrained.

**MIN_VOLUME** *real*

The minimum volume parameter is used to constrain the minimum cell (level 2, sampling-cell) volume. The default value is (MIN_LENGTH)$^3$. A non-positive value means the minimum volume is unconstrained.

**REF_TIME** $real_+$

The default reference time is REF_LENGTH / $|V_{\text{ref}}|$ .

**INITIAL_CONDS** [VACUUM | GAS_REF_CONDS | MANUAL]

This determines the method by which molecules are initialized in each level 2 cell.

**INITIAL_CONDS** VACUUM

No molecules are initialized in the flowfield. This type is only meaningful for the case where molecules will enter the simulation from one or more defined boundary conditions.

**INITIAL_CONDS** GAS_REF_CONDS

Molecules are initialized in each cell by sampling each required property assuming an equilibrium gas at the gas reference conditions.

**INITIAL_CONDS** MANUAL

This type allows the user to prescribe, independently, a spatial variation of properties, a variation in properties between species, and/or a variation (or non-equilibrium) between modes or energy components of each species.

The (optional) spatial variation is defined in terms of selected volumes XYZ_BOX (right parallelipipeds, or 'boxes') or selected level 1 cells (LVL1_CELL) in the domain. Properties are constant in all cells contained within a volume or level 1 cell. The entries are read in the order listed. In the case of an overlapping volume or repeated cell references, the final reference is used. Unassigned cells are set by interpolation between the nearest (two or three) distinct, surrounding entries or by the values at the nearest defined entry, if no surrounding entries can be identified. Each box or cell definition is followed by the description of the properties that exist therein. Lack of either type of definition implies no spatial variation in the domain.

**XYZ_BOX**

**XYZ_MIN** *real real real*

**XYZ_MAX** *real real real*

**LVL1_CELL** $int_+$

This line identifies the index of the level 1 cell.

Following the definition of each spatial subvolume (if any), the user must define the properties to be assigned to each species, and the method by which molecules are individually sampled. Any property that is not explicitly prescribed is assigned using GAS_REF_CONDS. Each species may be initialized with its own density, velocity, and temperature, analogous to GAS_REF_CONDS AMBIENT. Each basic property (or mode) of a molecule of a given species may be initialized independently, and may be sampled from an equilibrium distribution, or sampled from a discrete distribution function defined by additional input.

**SPECIES** *string*

This identifies the species for which distinct properties are to be defined.

**PROP** [VEL_[X|Y|Z] | ENERGY_TRANS | ENERGY-modename] METHOD [EQUILIB | DISCRETE]

This line identifies a property of the species and the method to be used to sample or initialize the property for each molecule. The available properties are the velocity components, the translational energy, and the energy content of a named internal mode (modename). Additional parameters are supplied on subsequent lines depending on the property and the method used. This line and its associated data is repeated for each property to be prescribed.

**PROP** VEL_[X|Y|Z] METHOD EQUILIB

The cartesian component of velocity is sampled from an equilibrium distribution parameterized by a bulk or drift velocity magnitude and a 'directed' translational temperature in this coordinate direction. The parameters are the velocity magnitude in this direction, and the temperature in this direction:

**VEL** *real* TEMP $real_+$

**PROP** VEL_[X|Y|Z] METHOD DISCRETE

The cartesian component $i$ of velocity, $v_i$, is sampled from an arbitrary discrete distribution function $f(v_i)$. This distribution is defined by the number of discrete partitions or bins into which this velocity space is subdivided, and the bin minimum (velocity) value and relative population. Note that this method cannot recover features of an arbitrary three-dimensional velocity distribution $f(v_x, v_y, v_z)$.

**NUM_BINS** $int_+$

**BIN** *real* POPULATION $real_+$

**PROP** ENERGY_TRANS METHOD [EQUILIB | DISCRETE]

This line allows the user to prescribe the translational energy magnitude. This item is not yet defined.

**PROP** ENERGY-modename METHOD [EQUILIB | DISCRETE]

This line allows the user to prescribe the energy content of the internal mode named modename.

**PROP** ENERGY-modename METHOD EQUILIB

The mode energy is sampled from an equilibrium distribution parameterized by the mode temperature:

**TEMP** $real_+$

**PROP** ENERGY-modename METHOD DISCRETE

The mode energy is sampled from an arbitrary discrete distribution function. This distribution is defined by the number of discrete partitions or bins into which energy space is subdivided, and the bin value and relative population. For modes of continuous energy distribution, the bin is defined by its minimum energy value. For quantized modes, the bin is defined by the bin level (integer).

**NUM_BINS** $int_+$

**BIN** [*real* | *int*] POPULATION $real_+$

**RANDOM_SEED** [$int_+$ | AUTO]

This defines the method by which the random number generator is seeded or initialized.

**RANDOM_SEED** AUTO

A seed is calculated for each processor based on the calendar date and time and the process number.

**RANDOM_SEED** $int_+$

The value supplied is used for each processor. (Note: the seed is typically a large odd integer).

#### 3.2.2.6 Grid Definition

SUPREM utilizes a 2-level generalized cartesian grid. The grid is defined and stored independently of the geometry. The level 1, or top-level grid is used mainly to decompose the domain for parallelization. This grid is static, or fixed, during a run. It is desirable to have the computational work per level 1 cell as uniform as possible throughout the domain. Options are provided to allow clustering of the level 1 grid in each coordinate direction, independently, to improve the basic grid structure for problems involving complex flowfields. Each level 1 cell can be independently subdivided into a relatively arbitrary number of uniformly-sized level 2 cells. The level 2 cells serve as the discretization of the flowfield in which DSMC sampling and collision pair selection is accomplished. The level 2 grid may be adapted during a simulation.

The Grid Definition section of the input file defines the level 1 and level 2 grid structure (the number of cells and their dimensions) and the spatial weight factors to be applied to the level 2 cells.

**LVL1_SPACING** [AUTO | MANUAL | LIST | REFINE]

**LVL1_SPACING** AUTO

The domain is decomposed into equal-sized level 1 cells. No additional parameters are required. The constraints are: number of cells...

**LVL1_SPACING** MANUAL

The non-NOVICE user can control the spacing of cells in each coordinate direction independently using one of several spacing models.

**DIRECTION** [X | Y | Z]

**NUM_CELLS** $int_+$

**DISTRIBUTION** [EQUAL_SIZE | TANH | GEOMETRIC | LIST | REFINE]

**DISTRIBUTION** EQUAL_SIZE

The domain is discretized into NUM_CELLS equal-sized cells in the given direction.

**DISTRIBUTION** TANH

The spacing is determined using a hyperbolic tangent function, which requires definition of the dimension of the first (at the minimum coordinate direction boundary) and last (at the maximum coordinate direction boundary) cell.

**DELTA_MIN_BDRY** $real_+$

**DELTA_MAX_BDRY** $real_+$

**DISTRIBUTION** GEOMETRIC

The spacing is determined using a geometric progression which requires definition of the dimension of the first (at the minimum coordinate direction boundary) and last (at the maximum coordinate direction boundary) cell.

**DELTA_MIN_BDRY** $real_+$

**DELTA_MAX_BDRY** $real_+$

**LVL1_SPACING** LIST

The spacing is defined by a list of vertices, one per line, beginning at the minimum boundary and ending at the maximum boundary of the coordinate direction.

**VERTEX** *real*

**LVL1_SPACING** REFINE

The spacing is generated by attempting to refine an existing discretization. The method will attempt to select a grid in which each level 1 cell has the same number of real molecules, based on the local densities in the existing grid. No additional parameters are required.

**LVL2_SPACING**

This is the header for the section which defines the discretization of level 1 cells into level 2 cells. In each level 1 cell, the level 2 cells are uniformly spaced.

**INITIAL** [AUTO | LIST | REFINE]

This section establishes the method used to calculate the initial level 2 discretization.

**INITIAL** AUTO

Each level 1 cell is subdivided independently based on the CELL_SIZE QUALITY parameter and the local or gas reference conditions in the level 1 cell. No additional parameters are required. The constraints are: number of cells...

**INITIAL** LIST

The non-NOVICE user can define the number of cells to be used in each direction in the level 1 cells contained in selected volumes XYZ_BOX (right parallelipipeds, or 'boxes') in the domain, or in selected level 1 cells (LVL1_CELL).

**XYZ_BOX**
**XYZ_MIN** *real real real*
**XYZ_MAX** *real real real*
**DIRECTION** [X | Y | Z]
**NUM_CELLS** $int_+$

**LVL1_CELL** $int_+$
**DIRECTION** [X | Y | Z]
**NUM_CELLS** $int_+$

**INITIAL** REFINE

The spacing is generated by attempting to refine an existing discretization. The method will attempt to select a level 2 spacing in each level 1 cell to satisfy the QUALITY CELL_SIZE constraints.

**ADAPTION** [OFF | AUTO | MANUAL ]

This section prescribes the methods for dynamic local adaption of the level 2 subgrid in each level 1 cell.

**ADAPTION** OFF

The initial level 2 grid is unaltered during the simulation.

**ADAPTION** AUTO

The level 2 spacing is adapted to attain the desired CELL_SIZE QUALITY. The constraints are xxxx.

ADAPTION MANUAL

The user defines the frequency at which the level 2 grid is evaluated, and the model used to adapt the grid. (See also QUALITY CELL_SIZE). This will be documented at a later date.

**DIAGNOSTICS** [OFF | MIN | MORE | MAX]

This controls the level of run-time diagnostics that are output to stdout during grid generation.

**LVL2_WEIGHT_FACTORS**

This is the header for the section which defines 'FNUM', the spatial weight factor (i.e., the number of real molecules represented by each simulated molecule) in each level 2 cell. These factors are typically adjusted locally to improve collision pair selection (facilitate nearest-neighbor collisions) or to reduce the number of simulated molecules (and thus, approximately, the amount of computational effort) in certain regions of a complex flowfield.

**INITIAL** [AUTO | MANUAL | LIST | REFINE]

This section establishes the initial weight factors.

INITIAL AUTO

FNUM is dependent on the cell dimension $\Delta$, the local (or reference) mean free path $\lambda$, and the dimensionality of the problem ($d = 1, 2, 3$). A value is assigned such that the number of simulated molecules initially in the cell is the greater of an empirical value ($4^d$) or $(\Delta/\lambda)^{1/3}$. The constraints are: number of cells...

INITIAL MANUAL

The non-NOVICE user can define the variation of FNUM over the domain in terms of selected analytical functions (constant over the domain, or based on cell volume, or to achieve constant flux across level 1 cell boundaries, or to minimize the number of simulated particles used during the unsteady portion of a steady flow simulation, etc.). This will be documented at a later date.

INITIAL LIST

The non-NOVICE user can define the weight factors to be used in selected volumes (XYZ_BOX) or in selected level 1 cells (LVL1_CELL). This is analogous to the procedure used in LVL2_SPACING INITIAL LIST (page 34).

**XYZ_BOX**

**XYZ_MIN** *real real real*
**XYZ_MAX** *real real real*
**FNUM** *real*$_+$
**LVL1_CELL** *int*$_+$
**FNUM** *real*$_+$

**INITIAL REFINE**

The spacing is generated by attempting to refine an existing weight factor distribution. The method will attempt to select the weight factor in each level 2 cell to satisfy the QUALITY COLLN_SEP constraints.

**ADAPTION** [OFF | AUTO | MANUAL ]

This section prescribes the methods for dynamic local adaption of the level 2 weight factors.

**ADAPTION OFF**

The initial spatial weight factors are maintained during the simulation.

**ADAPTION AUTO**

The level 2 weight factors are adapted to attain the desired QUALITY COLLN_SEP. The constraints are xxxx.

**ADAPTION MANUAL**

The user defines the frequency at which the weight factor values are evaluated, and the model used to adapt the factors. (See also QUALITY COLLN_SEP). This will be documented at a later date.

#### 3.2.2.7 Time Step Definition

The user must define the timestep to be used (time integration in the DSMC method is 'explicit'), and must define whether unsteady (transient) properties (*e.g.*, evolution of the flowfield in time from given initial conditions, or due to time-dependent boundary conditions) or steady properties (time-averaged conditions, with the time averaging initiated after transient phenomena have suitably decayed) are of interest.

**TIME_DEP** [STEADY | UNSTEADY]

The time dependence of the solution quantities of interest is either steady or unsteady. If UNSTEADY, only the UNSTEADY_PHASE subsection is required. If STEADY, both the UNSTEADY_PHASE and STEADY_PHASE subsections are required.

**TIME_STEP_CONTROL**

This is the header for the subsection providing for definition of the timestep, $\Delta t$. If this is an unsteady problem, then a single global value of $\Delta t$ is used to march the solution in physical time from the current time $t$ to the next time $t' = t + \Delta t$. For the solution to achieve time accuracy, this timestep must be small relative to the temporal phenomena to be resolved. If this is a steady problem, then the timestep can vary between cells (local time stepping) to improve solution efficiency, and in this case there is no meaningful physical time parameter.

**INITIAL** [AUTO | MANUAL | LIST | REFINE]

This subsection establishes the initial timestep.

**INITIAL** AUTO

For an unsteady problem, the timestep DT is set to one-half of the minimum local (or reference) collision frequency in the domain. For a steady problem, DT is set in each cell based on the TIME_STEP QUALITY parameter and the local or gas reference conditions in the cell. The constraints are ...

**INITIAL** MANUAL

The non-NOVICE user can define the variation of DT over the domain in terms of selected analytical functions (constant over the domain, or based on cell volume, or to achieve constant flux across level 1 cell boundaries, etc.). For an unsteady problem, the value may only be set to a single global constant. This will be documented at a later date.

**INITIAL** LIST

The non-NOVICE user can define the timesteps to be used in selected volumes (XYZ_BOX) or in selected level 1 cells (LVL1_CELL). This is analogous to the procedure used in LVL2_SPACING INITIAL LIST (page 34).

**XYZ_BOX**
  **XYZ_MIN** *real real real*
  **XYZ_MAX** *real real real*
    **DT** $real_+$
**LVL1_CELL** $int_+$
  **DT** $real_+$

**INITIAL** REFINE

The timestep is calculated by refinement of an existing timestep distribution. The method will attempt to select the timestep in each level 2 cell to satisfy the QUALITY TIMESTEP constraints. This is available only for a steady problem.

**ADAPTION** [OFF | AUTO | MANUAL ]

This section prescribes the methods for dynamic local adaption of the level 2 timesteps. This is available only for a steady problem.

**ADAPTION** OFF

The initial timesteps are maintained during the simulation.

**ADAPTION** AUTO

The timesteps are adapted to attain the desired QUALITY TIMESTEP. The constraints are xxxx.

**ADAPTION** MANUAL

The user defines the frequency at which the timesteps are tested, and the model used to adapt the timesteps. (See also QUALITY TIMESTEP). This will be documented at a later date.

**UNSTEADY_PHASE**

This is the header for the definition of control parameters to be applied during the unsteady portion of the simulation. The user can control the value to which the physical time parameter is initialized, the frequency at which the data structures containing sampled data are reinitialized (to zero), and the criteria used to determine the completion of the phase.

**INIT_TIME** *real*

This initializes the physical time parameter in the solution. This is intended to be used when an unsteady simulation is initialized via an existing steady solution.

**RESET_SAMPLE** [AUTO | MANUAL]

This determines the method by which all sampled data are reset (reinitialized to zero).

**RESET_SAMPLE** AUTO

The sampled data are reset at a constant timestep increment estimated from initial conditions. For a steady run, the increment is such to provide approximately four resets during the unsteady phase. For an unsteady run, the increment is every ten timesteps.

**RESET_SAMPLE** MANUAL

The user can input the number of timesteps between resets.

**NUM_STEPS** *int*$_+$

**COMPLETION_CRITERIA [AUTO | MANUAL]**

This determines the method by which the phase is considered to be completed.

**COMPLETION_CRITERIA AUTO**

The criteria is based on the temporal variation of the number of simulated molecules in the domain. ...This option is available only for steady cases.

**COMPLETION_CRITERIA MANUAL**

The user supplies the criteria from one of several methods.

[NUM_STEPS $int_+$ | FLOW_TIME $real_+$ | FLOW_REF_TIME $real_+$ | STD_DEV $real_+$ | WALL_CLOCK_HRS $real_+$]

**NUM_STEPS** $int_+$

The phase is complete when each cell has completed NUM_STEPS timesteps.

**FLOW_TIME** $real_+$

The phase is complete when each cell has completed a sufficient number of timesteps to achieve a cumulative cell 'integration' time ($\Sigma\Delta t$) $\geq$ FLOW_TIME units.

**FLOW_REF_TIME** $real_+$

The phase is complete when each cell has completed a sufficient number of timesteps to achieve a cumulative cell 'integration' time ($\Sigma\Delta t$) /REF_TIME $\geq$FLOW_REF_TIME.

**STD_DEV** $real_+$

The phase is completed when the temporal variation in the number of simulated molecules in the domain is ..... This will be documented at a later date.

**WALL_CLOCK_HRS** $real_+$

The phase is complete when the job has run for the given amount of wall clock hours (the difference between the current time and the job start time on the host process).

**STEADY_PHASE**

This is the header for the definition of control parameters to be applied during the steady portion of the simulation. The user can control only the criteria used to determine the completion of the phase. This section is ignored for UNSTEADY runs.

**COMPLETION_CRITERIA [AUTO | MANUAL]**

This determines the method by which the phase is considered to be completed.

**COMPLETION_CRITERIA AUTO**

The criteria is based on achieving the QUALITY CELL_SAMPLE_SIZE and BDRY_SAMPLE_SIZE requirements.

**COMPLETION_CRITERIA MANUAL**

The user supplies the criteria. The methods are the same as for UNSTEADY_PHASE, except that STD_DEV is not available.

[NUM_STEPS $int_+$ | FLOW_TIME $real_+$ | FLOW_REF_TIME $real_+$ | WALL_CLOCK_HRS $real_+$]

**DIAGNOSTICS** [OFF | MIN | MORE | MAX]

This controls the level of run-time diagnostics that are output to stdout during timestep calculation.

#### 3.2.2.8 Algorithm Control

This section allows the user to activate or deactivate several features of the basic DSMC algorithm.

**ALGORITHM_CONTROL**

This is the header for the section.

**SPECIES_WEIGHT_FACTORS** [OFF | AUTO | MANUAL]

This defines the relative weight factors to be used for each species in a multi-species simulation. These are used to alter the relative number of real molecules represented by each simulated molecule of a given species, and can assist in, e.g., resolving phenomena arising from trace species.

**SPECIES_WEIGHT_FACTORS** OFF

This means that all molecules have a weight of unity, regardless of species.

**SPECIES_WEIGHT_FACTORS** AUTO

This assigns the relative weight factors for each species based on the inverse of the relative populations defined in the GAS REF_CONDS.

**SPECIES_WEIGHT_FACTORS** MANUAL

The user supplies the weight factor for each species in the form:

**SPECIES** *string* WEIGHT $real_+$

Any species not listed will be assigned a weight factor of unity.

**<BACKGROUND_SPECIES** *string*>

This optional line defines the species named *string* as a background or passive species. This is available only for multi-species simulations. The objective of this option is to allow the use of decoupled or 'overlay' techniques to resolve features that are only weakly dependent on the background species. By default, all species (all molecules) are considered active and undergo basic movement and experience collision events. A molecule of a background species does not undergo movement, and only participates in collisions to a limited extent, depending on the collision process involved; a background molecule selected for collision can contribute its energies to the process, but it experiences no post-collision alteration or redistribution of energy. Further details are given in the Theory manual[]. There can be multiple instances of this line (but at least one species must remain active).

**COLLISIONS** [ON | OFF]

This line determines whether inter-molecular collisions are simulated (ON) or not (OFF).

**WEIGHT_FACTORS** [OFF | AUTO | MANUAL]

This defines the relative weight factors to be used to calculate the number of collision events that occur between various species in a multi-species simulation. These can assist in, e.g., resolving phenomena that arise from collisions involving trace species.

**WEIGHT_FACTORS** OFF

This means that all collision pairs have equal weight (a weight of unity), regardless of the species involved.

**WEIGHT_FACTORS** AUTO

This assigns the relative weight factors for each collision pair based on the product of the individual species weight factors.

**WEIGHT_FACTORS** MANUAL

The user supplies the weight factor for each (ordered) species pair in the form:

**COLLN_PAIR** *string string* WEIGHT $real_+$

Any species pair not listed will be assigned a weight factor of unity.

**DIAGNOSTICS** [OFF | MIN | MORE | MAX]

This controls the level of run-time diagnostics that are output to stdout during the collision phase.

**MOVEMENT** [ON | OFF]

This line determines whether molecule movement is simulated (ON) or not (OFF).

**EXTERNAL_FORCE** [OFF | GRAVITY *real real real*]

This line determines whether a molecule experiences an external or body force during movement. A value of OFF means the molecule trajectory during movement is a straight line in the direction of the velocity vector, $\Delta\vec{r} = \vec{v}\Delta t$.

**EXTERNAL_FORCE** GRAVITY *real real real*

A constant acceleration (force per unit mass), defined by the three cartesian components of the acceleration, is applied during movement. The parabolic trajectory is approximated as xxxxx.

**DIAGNOSTICS** [OFF | MIN | MORE | MAX]

This controls the level of run-time diagnostics that are output to stdout during the movement phase.

**SAMPLE_CELL** [ON | OFF]

This line activates (ON) or deactivates (OFF) basic sampling of properties in all cells in the domain. The value of OFF is not recommended. The basic cell sampling includes each species mean mass, momentum (i.e., velocity, three cartesian coordinates), square of the momenta (i.e., directed translational energy, three cartesian coordinates) and internal energies. Sampling of additional types of quantities in specific regions or volumes may be initiated with the remaining optional lines of this section. There can be multiple occurrences of the location and associated type definitions.

<[**XYZ_LIST** *string* | **XYZ_BOX** *string* | LVL1_LIST *string*]>

This optional line selects the method used to describe the volumes for which additional sampling is done. This can be a list of positions, regular volumes, or cells. The string is a user-supplied name used only to reference this sampling definition for output (WRITE SAMPLE_LIST, page 47). If this line does not appear, then any TYPE line that follows will be applied to all cells in the domain.

**XYZ_LIST** *string*

Additional sampling is done in the cell whose centroid is nearest the supplied coordinates. The coordinates are entered one per line:

**XYZ** *real real real*

**XYZ_BOX** *string*

Additional sampling is done in all cells which lie in or on the 'box' (volume) defined by the minimum and maximum vertices:

**XYZ_MIN** *real real real*

**XYZ_MAX** *real real real*

**LVL1_LIST** *string*

**CELL** $int_+$

**TYPE** [DETAIL | MACRO | INTEGRATED | MOL_LIST]

This line indicates the additional properties which are to be sampled (and made available for output).

**TYPE** DETAIL

This indicates that the distribution function of certain pre-defined scalar properties of specific species is to be sampled. The properties and the species are defined in the species definition section of the gas file, the user must supply the name of the SAMPLE definition (see Section 3.4.2.1). This is intended to allow the user to analyze the flowfield at a more 'detailed' or 'microscopic' level.

**SAMPLE** *string*

**TYPE** MACRO

This indicates that certain pre-defined macroscopic properties derived from the default sampled data are of interest. This is intended to allow the user to easily extract specific quantities of interest for specific species directly, rather than through the postprocessor. The species and properties are defined with the following SPECIES line. There can be multiple occurrences of the SPECIES line.

**SPECIES** [EACH | MASS_AVG | *string*] PROP [NUM_DENSITY | MASS_DENSITY | VEL_X | VEL_Y | VEL_Z | PRESSURE | TEMP | TEMP_X | TEMP_Y | TEMP_Z | TEMP-modename]

**TYPE** INTEGRATED

This indicates that the value $V$ of some quantity $v$ integrated over the defined spatial region (and over all species) is to be calculated ($V = \Sigma_{XYZ} v(x, y, z)$). This is intended to allow the user to easily extract integrated quantities in specific volumes. The available properties are:

**PROP** [MASS | MOMENTUM_X | MOMENTUM_Y | MOMENTUM_Z | ENERGY_TRANS | ENERGY_INTERNAL]

**TYPE** MOL_LIST

This indicates that the sampling will store a list of molecules that reside in the sampling volume. Each entry in the list includes the complete molecule description. The user must identify the species and the number of molecules that are to be stored in the list. (Once the list is filled, entries are overwritten as necessary in first-in-first-out order).

**SPECIES** *string*

**NUMBER** $int_+$

**SAMPLE_COLLN** [ON | OFF]

This line activates (ON) or deactivates (OFF) basic sampling of collision properties in all cells in the domain. The value of OFF is not recommended. The basic collision sampling includes each species-pair selection and acceptance counts, separation distance, and each process acceptance count, cross section and collision energy. (All data except the selection count involve only those pairs that have been accepted for collision). Sampling of additional types of quantities in specific regions can be initiated with the remaining optional lines of this section. There can be multiple occurrences of the location and associated type definitions.

<[**XYZ_LIST** *string* | **XYZ_BOX** *string* | LVL1_LIST *string*]>

The sampling volume definitions are the same as for SAMPLE_CELL.

**TYPE** [DETAIL | MACRO | INTEGRATED | MOL_LIST]

This line indicates the additional properties which are to be sampled (and made available for output).

**TYPE** DETAIL

This indicates that the distribution function of certain pre-defined scalar properties of a specific species in a collision pair is to be sampled. The properties and the species are defined in the bimolecular collision definition section of the gas file, the user must supply the name of the SAMPLE definition (see Section 3.5.2). This is intended to allow the user to analyze specific collision processes at a more 'microscopic' level.

**SAMPLE** *string*

**TYPE** MACRO

This indicates that certain pre-defined macroscopic properties derived from the default sampled collision-pair data are of interest. This is intended to allow the user to easily extract specific quantities for specific species. The pair and properties are defined with the following COLLN_PAIR line. There can be multiple occurrences of the COLLN_PAIR line.

**COLLN_PAIR** *string string* PROP [FREQ | RATE | REL_SPEED | SIGMA_SPEED | ENERGY_TRANS | E_COLLN | ]

**TYPE** MOL_LIST

This indicates that the sampling will store a list of pairs of molecules that have been accepted for collision (note: the nature of the acceptance procedure means that these properties will not necessarily be equivalent to or representative of those of pairs selected at random). The properties stored are those of the pre-collision state. The user must identify the species of the collision pair and the number of pairs of molecules that are to be stored in the list.

**COLLN_PAIR** *string string*

**NUMBER** $int_+$

**SAMPLE_BDRY** [ON | OFF]

This line activates (ON) or deactivates (OFF) basic sampling of the properties at all boundary elements in the domain. The value of OFF is not recommended. The basic boundary sampling includes the species-resolved directed flux of mass, momentum (velocity), and energy (translational and internal). The fluxes are differentiated as incident (or incoming) to, and outgoing (or reflected) from, the boundary. The momenta and the translational energy components are decomposed into the element-based coordinate system (normal and two tangential coordinates). Sampling of additional types of quantities at specific boundaries can be initiated with the remaining optional lines of this section. There can be multiple occurrences of the location and associated type definitions.

<[**XYZ_LIST** *string* | **ENTITY_LIST** *string* ]>

This optional line selects a list of positions or named geometrical entities or components for which additional sampling is done. The string is a user-supplied name used only to reference this sampling definition.

**XYZ_LIST** *string*

This activates sampling on the boundary element whose centroid is nearest each XYZ entry.

**XYZ** *real real real*

**ENTITY_LIST** *string*

This activates sampling on all elements of the geometrical entity or component named in the following identification statement. There can be multiple occurrences of this statement:

**ENTITY** *string* <COMPONENT *string* >

**TYPE** [DETAIL | MACRO | INTEGRATED | MOL_LIST]

This line indicates the additional properties which are to be sampled (and made available for output).

**TYPE** DETAIL

This indicates that the distribution function of certain pre-defined scalar properties of specific species which interact with the boundary is to be sampled. The properties and the species are defined in the interaction definition section of the boundary condition file, the user must supply the name of the SAMPLE definition (see Section 3.5.1). This is intended to allow the user to analyze the properties of species present at a boundary at a more 'microscopic' level.

**SAMPLE** *string*

**TYPE** MACRO

This indicates that certain pre-defined macroscopic properties derived from the default sampled data are of interest. This is intended to allow the user to easily extract specific quantities of interest for specific species directly, rather than through the postprocessor. The species and properties are defined with the following SPECIES line. There can be multiple occurrences of the SPECIES line. The available properties are the fluxes (transported property per unit area and per unit time) of mass, and energy, and the pressure (normal momentum flux) and shear (tangential momentum flux, two components).

**SPECIES** [EACH | MASS_AVG | *string*] PROP [MASS_FLUX_[INC | OUT | NET]| PRESSURE | SHEAR | ENERGY_FLUX_[INC | OUT | NET]]

**TYPE** INTEGRATED

This indicates that the value $F$ of some fluxal quantity $f$ integrated over the defined boundary area is to be calculated ($F = \Sigma_{\text{area}} f(x, y, z)$). This is intended to allow the user to easily extract integrated quantities on specific entities or components. The available properties are the INComing (or INCident), OUTgoing (or reflected), or net (INC - OUT) mass, momentum, and translational energy transport per unit time, and the net forces and moments: MASS_[INC | OUT | NET], MOMENTUM_[X | Y | Z]_[INC | OUT | NET], ENERGY_TRANS_[INC | OUT | NET], FORCE_[X | Y | Z], MOMENT_[X | Y | Z].

**TYPE** MOL_LIST

This indicates that the sampling will store a list of molecules incident on this boundary (from the interior). The properties stored are those of the incident state. The user must identify the species and the number of molecules which are to be stored.

**SPECIES** *string*

**NUMBER** $int_+$

#### 3.2.2.9 Output Control

This section controls the generation of output files produced by the code. There are three types of files: standard output (and standard error); restart files containing cell, molecule, and boundary element data; and various optional output files produced by user-defined sampling. Filenames are of the form 'prefix.suffix', and all files produced by a single run of the code will possess the same prefix.

**OUTPUT** [ON | OFF ]

This line activates (ON) or deactivates (OFF) the generation of output files by the code. The value of OFF produces only the standard output and is not recommended.

**PREFIX** *string*

This defines the prefix by which all files produced during this run are identified.

The remaining entries in this section define the method by which each type of output file is written. The general form is:

**WRITE** [STDOUT | RESTART | SAMPLE_LIST] [OFF | AUTO | MANUAL]

This line is followed by required parameters for the MANUAL option. In this case, the user can control the format of each type of file, the frequency it is written, and whether the files are over-written on subsequent write operations.

**WRITE** STDOUT [OFF | AUTO | MANUAL]

Standard output contains informational and warning messages from the code, a brief (one or two line) running summary of the progress of the simulation (updated at a given frequency), and formal output produced upon completion. The suffix for the file is 'out'. This file is always written in formatted form, and each subsequent entry is always appended to the file.

**WRITE** STDOUT OFF

This option is not meaningful.

**WRITE** STDOUT AUTO

This option means the running summary is updated with a frequency dependent on the global timestep number for the current phase (unsteady or steady): step 1,2,3,...,10,20,..., 100, 200, ...

**WRITE** STDOUT MANUAL

This option allows the user to define the frequency with which the running summary of the simulation is updated. This frequency can be given by a fixed increment of global timesteps, or at a fixed increment of wall clock hours.

**FREQ** [NUM_STEPS $int_+$ | WALL_CLOCK_HRS $real_+$]

**WRITE** RESTART [OFF | AUTO | MANUAL]

Restart output contains all the data required to subsequently restart, refine, or post-process a run. There are three restart files: boundary element information (suffix 'bdry'), containing datastructures defining the boundaries of the domain and the properties of each boundary element; cell information (suffix 'cell') containing the cell datastructures and basic simulation parameters; and molecule information (suffix '.mol') containing the properties of each molecule. These files can be very large for a complex simulation, and in general size(bdry) $\ll$ size(cell) $\ll$ size(mol). Nominally, post-processing utilizes the cell and boundary files, while the molecule file need be retained only for subsequent restart runs or more detailed analysis.

**WRITE** RESTART OFF

This option means that no restart files are produced. This option is not recommended.

**WRITE** RESTART AUTO

This option selects default values for the format of the restart files and the frequency with which they are written; the files are overwritten by subsequent restart output operations. The defaults are:

**FORMAT** UNFORM
**FREQ** WALL_CLOCK_HRS 2.0
**OVERWRITE** TRUE

**WRITE** RESTART MANUAL

This option allows the user to select the format, frequency, and the type of write operation used to write the restart files.

**FORMAT** [FORM | UNFORM]

The files can be written in fortran FORMatted (ascii) or UNFORMatted form.

**FREQ** [NUM_STEPS $int_+$ | WALL_CLOCK_HRS $real_+$]
**OVERWRITE** [TRUE | FALSE]

The operation of writing the restart files consists of opening the file, writing the data, and closing the file. The OVERWRITE TRUE option indicates that the files will be opened with the fortran directive "position=rewind", and any previously-written restart data will be deleted, while FALSE means that the files will be opened with fortran "position=append". The option FALSE may be useful for unsteady runs.

<WRITE SAMPLE_LIST [OFF | AUTO | MANUAL]>

Sample list output contains the data relating to optional, user-defined cell, collision, or boundary sampling. Each SAMPLE entry can produce one output file, with the file suffix being the first three characters of the SAMPLE list name. (An integer based on the order of appearance of an entry in the input files is appended if necessary to assure filename uniqueness). These files are intended to be used for simple spread-sheet analysis or x-y plotting.

**WRITE** SAMPLE_LIST OFF

This option means that sample files are not produced. The data generated by the sample definition will be present in the appropriate restart file (bdry or cell), however.

**WRITE** SAMPLE_LIST AUTO

This option selects default values for the format of all sample list files and the frequency with which they are written; the data are overwritten by subsequent sample list output operations. The values are:

**FORMAT** FORM
**FREQ** WALL_CLOCK_HRS 0.25
**OVERWRITE** TRUE

**WRITE** SAMPLE_LIST MANUAL

This option allows the user to select the format, frequency, and the type of write operation used to write each sample list dataset. The name of the sample list must be supplied. There can be one instance of this section for each sample list, if a list does not appear, the AUTO values are used for it.

**NAME** *string*

This provides the list name as it appears in the definition of the sample list.

**FORMAT** [FORM | UNFORM]
**FREQ** [NUM_STEPS $int_+$ | WALL_CLOCK_HRS $real_+$]
**OVERWRITE** [TRUE | FALSE]

## 3.3 Geometry File

The geometry file provides the definition of all boundaries in the domain (including the domain boundaries themselves). The domain can consist of an arbitrary number of boundaries. The user can also define virtual surfaces which allow for the tracking or filtering of molecules which are incident upon the surface. A set of geometric primitives are provided to allow analytical definition of relatively simple geometrical shapes in two or three dimensions. For more complex geometries, the user will be required to provide a surface triangulation from a CAD or grid generator.

### 3.3.1 Geometry File Template

### 3.3.2 Keyword Definition

## 3.4 Gas File

The gas file consists of two major sections. The first section defines all species that are to be included in the simulation. The second section defines the processes or reaction events these species can undergo during inter-molecular collisions. These processes are typically bi-molecular collisions, but allowance is made for tri-molecular (recombination) events and even for uni-molecular (e.g., spontaneous radiative decay) events.

The species definition consists of the species name and its elemental composition, followed by a subsection describing any internal energy "modes" that are to be defined. This subsection lists the number of modes, and the name and representation (continuous or quantized) of each.

The collisional interaction section (note: somewhat of a misnomer, since unimolecular processes are included) lists the species names involved in each interaction, and the processes which the species can undergo. The processes available are dependent on the number and type of species involved. Each process is described via a 'chemical equation' and associated keywords which define the model to be used and any required model-dependent parameters.

Gas files corresponding to several typical analysis scenarios are supplied. If these are not suitable, the user can modify one of these files or create a new gas file as necessary.

### 3.4.1 Gas File Template

```
#### SUPREM Version 1.0 Gas File ###
user-supplied file description string(s)
SPECIES_NAME string
  COMPOSITION string-int+ [string-int+...]
  NUM_INTL_MODES int
  MODE_NAME [string | RIGIDROT | SHOVIB | AHOVIB | LUMPED]
    mode-representation-dependent parameters
 <MODE_NAME, etc. repeat for 2..NUM_INTL_MODES>
<SAMPLE string PROP string [AUTO | MIN_BIN real+ MAX_BIN real+ NUM_BINS int+]>
<SAMPLE line repeats for remaining properties>
<SPECIES_NAME, etc. repeat for remaining species>
BIMOLECULAR [TRUE | FALSE]
  COLLN_PAIR string1 string2
    string1<()> + string2<()> → string3<()> + string4<()> <+ string5>
    PROCESS [ELASTIC | INELASTIC | REACTION]  MODEL name NUM_PARAMS int+
      model-dependent parameters (NUM_PARAMS lines)
     <COMMENT string>
     <SAMPLE string SPECIES_ID int+ MODE_NAME string>
     <SAMPLE line repeats for remaining modes>
    <reaction equation, PROCESS, etc., repeat for remaining processes>
```

```
  <COLLN_PAIR, etc. repeat for remaining collision pairs>
TRIMOLECULAR [TRUE | FALSE]
  COLLN_TRIPLE string1 string2 string3
    string1 + string2 + string3 → string4 + string5
    PROCESS RECOMB MODEL name  NUM_PARAMS int+
      model-dependent parameters (NUM_PARAMS lines)
     <COMMENT string>
   <reaction equation, PROCESS, etc. repeat for remaining processes>
 <COLLN_TRIPLE, etc. repeat for remaining collision pairs>
UNIMOLECULAR [TRUE | FALSE]
  PAIR string1 string2
    string1 → string2 + PHOTON
    PROCESS DEEXCITATION MODEL name  NUM_PARAMS int+
      model-dependent parameters (NUM_PARAMS lines)
     <COMMENT string>
    <deexcitation equation, PROCESS, etc. repeat>
    string1 + PHOTON → string2
    PROCESS EXCITATION MODEL name   NUM_PARAMS int+
      model-dependent parameters (NUM_PARAMS lines)
     <COMMENT string>
    <excitation equation, PROCESS, etc. repeat>
    <PAIR etc. repeat>
#### SUPREM Version 1.0 Gas File end ###
```

## 3.4.2 Keyword Definition

### 3.4.2.1 Species Definition

**SPECIES_NAME** *string*

Each species must have an arbitrary, unique name *string*. This name need not be the chemical composition of the species. The name must not contain a left or right parenthesis '(' or ')'.

**COMPOSITION** *string-int*$_+$ [*string-int*$_+$ ...]

The composition is the elemental composition of the species, in the form of an 'element name - number of atoms' pair, repeated for each unique element. The element name must be the standard 1- or 2-character chemical symbol for the element. For example, monatomic argon is defined as Ar-1, diatomic nitrogen as N-2, triatomic silicon dioxide as Si-1 O-2.

**NUM_INTL_MODES** *int*

This defines the number of internal modes to be modelled for the species. An "internal mode" is defined to be a component of non-translational energy that is present in a molecule of the species (all molecules possess a translational energy). Each species can possess an arbitrary number of internal energy components.

Each mode as defined here is distinct (decoupled) from all other modes, and must be describable by a single scalar value (an energy or a quantum level

for that mode) stored for the molecule. Each mode can potentially interact (undergo redistribution of energy) with translational energy or another mode during a collision, or even change its state via a unimolecular process (e.g., radiative deexcitation).

**MODE_NAME** [*string* | RIGIDROT | SHOVIB | AHOVIB | LUMPED]

The modes are identified by name. This can be an arbitrary user-defined *string*, or be a pre-defined type. The pre-defined types apply to certain common modes, and imply additional features about how the mode is modeled. The pre-defined modes are:

**RIGIDROT,** a quantized, bounded rigid rotor. This mode requires definition of THETA_ROT and MAX_ENERGY.

**SHOVIB,** a quantized, bounded simple harmonic oscillator vibrator. This mode requires definition of THETA_VIB and MAX_ENERGY.

**AHOVIB,** a quantized, bounded anharmonic oscillator vibrator defined by a Dunham series expansion. This mode requires definition of NUM_DUNHAM_TERMS, DUNHAM_COEFFS, and MAX_ENERGY.

**LUMPED,** one or more 'real' modes are lumped into a single fictitious, continuous, non-bounded mode (parameterized by an effective 'degrees of freedom'). This requires definition of DOF.

**MODE_REP** [CONTIN | QUANTIZ]

The energy content of each user-defined mode may be represented as a continuous or quantized variable. Additional parameters must then be supplied to completely define the mode.

**MODE_REP** CONTIN

**DOF** $real_+$

This defines the number of degrees of freedom.

**MAX_ENERGY** $real_+$

This defines the maximum available energy of the mode. This is measured relative to the ground state of the mode (which is assumed to be at zero energy).

**MODE_REP** QUANTIZ

**NUM_QUANTIZ_LVLS** $int_+$

This defines the number of quantized levels available. This can be an arbitrary subset of the levels available in the true mode, so that multiple levels of a single mode of a species can be modelled instead as distinct single-level modes of separate fictitious species.

**LVL** *int* DEGEN $int_+$ ENERGY $real_+$

This triplet lists the level number, the level degeneracy, and the level energy. This line is repeated for each of the NUM_QUANTIZ_LVLS levels. The levels must be input in order of increasing energy.

**MAX_ENERGY** *real*$_+$

This defines the maximum available energy of the mode.

The parameters for the pre-defined modes are:

**MODE_NAME** RIGIDROT

The energy of rotational level $j$ is defined by the rigid-rotor model, $e_j = j(j+1)k_B\Theta_r$, $j = 0, 1, \ldots j\text{max}$. The maximum allowable level $j\text{max}$ is found as the largest $j$ for which $e_j \leq$MAX_ENERGY.

[**MODE_REP** QUANTIZ]

**THETA_ROT** *real*$_+$

This defines the characteristic rotational temperature of the mode, $\Theta_r$.

**MAX_ENERGY** *real*$_+$

**MODE_NAME** SHOVIB

The energy of vibrational level $v$ is defined by simple harmonic oscillator model, $e_v = k_B\Theta_v(v + 1/2) - k_B\Theta_v/2, v = 0, 1, \ldots, v\text{max}$. Note that the energy levels have been offset by the zero-point energy such that $e_0 = 0$. The maximum allowable level $v\text{max}$ is found as the largest $v$ for which $e_v \leq$MAX_ENERGY.

[**MODE_REP** QUANTIZ]

**THETA_VIB** *real*$_+$

This defines the characteristic vibrational temperature of the mode, $\theta_v$.

**MAX_ENERGY** *real*$_+$

**MODE_NAME** AHOVIB

The energy of vibrational level $v$ is defined by the Dunham series expansion, $e_v = \sum_{k=1}^{n}(-1)^{k-1}\omega_k(v + 1/2)^k - e_o$. Note that the energy levels have been offset by the zero-point energy $e_o = e_{v=0}$. The maximum allowable level $v\text{max}$ is found as the largest $v$ for which $e_v \leq$MAX_ENERGY.

[**MODE_REP** QUANTIZ]

**NUM_DUNHAM_TERMS** *int*$_+$

This defines the number of terms $n$ in the expansion.

**DUNHAM_COEFFS** *real*$_+$ <*real*$_+$ ...>

This line lists the $n$ coefficients $\omega_k$ in order, $k = 1 \ldots n$.

**MAX_ENERGY** *real*$_+$

**MODE_NAME** LUMPED

[**MODE_REP** CONTIN]

**DOF** *real*$_+$

<**SAMPLE** *string* **PROP** name [**AUTO** | **MIN_BIN** *real*$_+$ **MAX_BIN** *real*$_+$ **NUM_BINS** *int*$_+$]>

This optional line indicates that storage is to be allocated in the sampling datastructures to allow sampling of the distribution of the named scalar property of the species. The *string* following the SAMPLE keyword is a user-supplied name used only to reference this sample definition (see SAMPLE_CELL TYPE DETAIL, Section 3.2.2.8). The property to be sampled is identified by keyword, and the remaining parameters define the resolution of the sampling. There can be multiple occurrences of the SAMPLE line per species.

**PROP** [VEL_X | VEL_Y | VEL_Z | ENERGY_TRANS | ENERGY-modename]

**VEL_X**, VEL_Y, VEL_Z

These are the components of velocity in the coordinate directions

**ENERGY_TRANS**

This is the magnitude of the translational energy

**ENERGY-modename**

This is the magnitude of the energy of MODE_NAME modename.

[**AUTO** | **MIN_BIN** *int*$_+$ **MAX_BIN** *int*$_+$ **NUM_BINS** *int*$_+$]

**AUTO**

This instructs the code to automatically calculate a default sampling range and resolution for the distribution.

**MIN_BIN** *real*$_+$ **MAX_BIN** *real*$_+$ **NUM_BINS** *int*$_+$

This instructs the code to sample over the range MIN_BIN to MAX_BIN, inclusive, using NUM_BINS equally spaced sampling bins.

#### 3.4.2.2 Bimolecular Collision Event Definition

**BIMOLECULAR** [TRUE | FALSE]

This line denotes the beginning of the bimolecular collision definition section. Setting it to FALSE means that the section will be ignored. A bimolecular collision involves two pre-collision molecules.

**COLLN_PAIR** *string1 string2*

This line lists the names of the species for which one or more binary collision equations are to be supplied. The ordering of the names must correspond to the order in which they occur in the pre-collision portion of the collision equation. The keyword ANY can be used for *string2* to denote that the definitions hold for any secondary species. The remainder of the subsection for the given pair consists of an arbitrary number of collision equations, each with an associated PROCESS keyword definition, and an optional SAMPLING keyword. Taken together, the collision equation and PROCESS entries define:

1. each interaction which is allowed to occur between modes in the species (not listed means the interaction cannot and will not occur).

2. which modes from which species contribute to the 'collision energy' or 'available energy' $e_c$.

3. the type of process involved, and the model which will be used to determine the process cross section.

4. the method by which the available energy is redistributed to the post-collision species, to their internal modes, and to the relative translational energy of the species.

Collisions are classified into three basic types (PROCESSES):

1. ELASTIC, allowing only the redistribution of the relative translational energy of the pair ($e_c = e_t$), regardless of the internal energies of the pair.

2. INELASTIC, allowing only the redistribution of the collision energy composed of the relative translational energy and the energy in a single mode of the primary (first-listed) molecule of the pair.

3. REACTION, generalizes (2) to allow the collision energy to be made up of an arbitrary combination of the energies of the pre-collision molecules, and allows for arbitrary redistribution of the energy into post-collision or new product species. The number of post-collision molecules may differ from the number of pre-collision molecules by unity or zero.

For the case where several PROCESSes are defined for a given collision pair, at most one process will occur. The process which occurs is determined by the relative cross section of that process using the null-collision scheme. A more detailed description of the processes, the models, and their implementation is provided in the Theory manual[].

The collision equation is:

*string1*<()> + *string2*<()> → *string3*<()> + *string4*<()> <+ *string5*>

This line defines an abstract collision process or reaction equation in the form pre-collision species_name(modenames) → post-collision species_name(modenames). This representation assumes that the process cross section or probability depends only on the translational energy and zero or more mode energies which are listed. The pre-collision modes that contribute to the collision energy are listed for each molecule in the form (modename_i, modename_j, ...), the left-to-right order in which the modes are listed is the order in which the collision energy is incremented (i.e., $e_c = e_t + e_i + e_j + \ldots$). Post-collision modes which undergo redistribution or reinitialization are denoted by a prime, e.g., (mode_name1'). The order in which the molecules are listed, and the order in which the modes are listed, is the order in which the molecules and the modes are redistributed. Relative translational energy is the final quantity redistributed.

**PROCESS [ELASTIC | INELASTIC | REACTION] MODEL name NUM_PARAMS** $int_+$

This line defines the process type, model, and model parameters corresponding to the collision equation. For each PROCESS type, there are one or more pre-defined models, each with a pre-defined number of named parameters. At present, these parameters are restricted to be real-valued scalar constants. Additional models can be added by definition of a model name, the number of parameters required, and the name of each parameter.

**PROCESS ELASTIC MODEL** [VHS | VSS | TABULATED] **NUM_PARAMS** $int_+$

An elastic process can utilize an analytical model (the VHS model of Bird or the VSS model of Koura) or can be a tabulated function of the relative translational energy.

**PROCESS ELASTIC MODEL VHS NUM_PARAMS 3**

The VHS elastic cross section is $\sigma_t(e_t) = \pi d^2$. The model requires a minimum of three parameters. The angular scattering model is isotropic.

**DIAM_REF** $real_+$

This is the collision reference diameter $d_{\text{ref}}$ at the reference temperature.

**VISC_EXPON** $real_+$

This is the exponent $\omega_b$ of the temperature-dependence of viscosity, $\mu(T) \sim T^{\omega_b}$.

**TEMP_REF** $real_+$

This is the reference temperature $T_{\text{ref}}$.

**[COMMENT** *string*]

This is an optional comment.

the post-collision energy of the mode is sampled from the available collision energy. The remaining energy is assigned to the relative translational energy of the pair. A reference temperature for which the collision number was evaluated is also required.

**COLLN_NUMBER** *real*$_+$

This is the constant collision number $Z_m$. Typically, $Z_m \gg 1$.

**TEMP_REF** *real*$_+$

This is a reference temperature $T_{\text{ref}}$ at which $Z_m$ was evaluated. This is used solely to evaluate the applicability of the model as applied to this process.

**[COMMENT** *string*]

This is an optional comment.

**PROCESS** REACTION **MODEL [TCE] NUM_PARAMS** *int*$_+$

A reaction process can currently utilize the Total Collision Energy (TCE) model of Bird. Additional models can be added by the user.

**PROCESS REACTION MODEL** TCE **NUM_PARAMS** 5

The Total Collision Energy (TCE) model of Bird calculates the reaction cross section based on a given elastic cross section and parameters of the Arrhenius rate coefficient equation $k(T) = \Lambda T^{\eta} \exp(-E_a/k_B T)$.

**COEFF** *real*$_+$

This is the constant coefficient $\Lambda$.

**TEMP_EXPON** *real*$_+$

This is the temperature exponent $\eta$.

**ENERGY_ACTIV** *real*$_+$

This is the activation energy $E_a$.

**HEAT_OF_REACT** *real*

This is the heat of reaction (*product* − *reactant*).

**CALIB_FACTOR** *real*$_+$

This is a constant steric or calibration factor which may be used to adjust the cross section to recover the expected rate coefficient.

**[COMMENT** *string*]

This is an optional comment.

**<SAMPLE** *string* **SPECIES_ID** $int_+$ **PROP** name [**AUTO** | **MIN_BIN** $real_+$ **MAX_BIN** $real_+$ **NUM_BINS** $int_+$]>

This optional line activates sampling of the distribution of the named scalar property of a particular species involved in this process. This line is analogous to the species SAMPLE line (see Section 3.4.2.1). The *string* following the SAMPLE keyword is a user-supplied name used only to reference this sample definition(see SAMPLE_COLLN TYPE DETAIL, Section 3.2.2.8). The species to be sampled is identified by the index of occurrence of the species in the collision equation (SPECIES_ID is 1 or 2 for the first or second pre-collision species, 3 or greater for the post-collision species). The property to be sampled is identified by its PROPerty name. The sampling range and resolution are determined by the remaining parameters. There can be multiple occurrences of the SAMPLE line per process. Available properties are currently those for the species SAMPLE line plus the following:

**PROP** [VEL_X | VEL_Y | VEL_Z | ENERGY_TRANS | ENERGY-modename | ENERGY_COLLN]

**ENERGY_COLLN** this is the process collision energy $e_c$.

### 3.4.2.3 Trimolecular Collision Event Definition

**TRIMOLECULAR** [TRUE | FALSE]

This line denotes the beginning of the trimolecular collision definition section. Setting it to FALSE means that the section will be ignored.

**COLLN_TRIPLE** *string1 string2 string3*

This line lists the names of the species for which ternary (recombination) collision equations are to be supplied. The ordering of the names must correspond to the order in which they occur in the pre-collision portion of the collision equation. There must be a corresponding BIMOLECULAR, COLLN_PAIR definition for *string1* and *string2*, and thus this section serves solely to identify an additional possible recombination process for the pair, given the third body species *string3* (keyword ANY is allowed). The process definition is analogous to that for COLLN_PAIR, and only an abbreviated description is supplied:

**PROCESS RECOMB MODEL [TCE] NUM_PARAMS** $int_+$

A reaction process can currently utilize the Total Collision Energy (TCE) model of Bird. Additional models can be added by the user. This will be documented at a later date.

#### 3.4.2.4 Unimolecular Event Definition

**UNIMOLECULAR** [TRUE | FALSE]

This line denotes the beginning of the unimolecular process definition section. Setting it to FALSE means that the section will be ignored. The section is intended to nominally allow the definition of discrete radiative deexcitation events which arise from quantized modes of a molecule. A provision is made for the reverse, excitation, event. The unimolecular feature is intended for future implementation.

**PAIR** *string1 string2*

This line lists the names of the species which are coupled via a radiative process. The species names can differ, implying that a true quantized mode has been partitioned into single-level modes of two or more fictitious species. The mode names are described on the radiative equation line analogous to the collision equation form. The specific form is dependent on whether it is a deexcitation or excitation process.

For DEEXCITATION:

*string1(modename1<=lvl1>)* $\rightarrow$ *string2(modename2<=lvl2>)* + PHOTON

If *string1* and *string2* are equivalent, then *modename1* and *modename2* must be equivalent, the mode must possess two or more levels, and the energy of level *lvl1* must be greater than that of *lvl2*. Otherwise, the species names are distinct, implying that the *modenames* are sufficient to uniquely define the event.

**PROCESS** DEEXCITATION MODEL name NUM_PARAMS *int*$_+$

The event leads to the creation of a PHOTON with energy equal to the energy difference between the pre- and post-event mode energies. The tentative model will typically require a single parameter corresponding to the radiative lifetime of the excited (pre-event) state.

## 3.5 Boundary Condition File

The boundary condition file defines the boundary conditions to be applied to the surfaces defined in the geometry file (Section 3.3). The boundary conditions prescribe the interaction which occurs when a molecule trajectory intersects a boundary element during the movement phase, and the means by which molecules may be spontaneously created at the boundary. A variety of BC types are available. Each type typically has several subtypes which differ in the model and parameters utilized. The typical application will involve steady, or time-independent boundary conditions; in this case, only one boundary condition may be defined for each surface. Unsteady or time-dependent boundary conditions may be created by defining multiple discrete conditions on a surface as described below.

The basic boundary condition types are summarized as:

1. SOURCE - an 'open' boundary at which incoming molecules are created. Molecules reaching the boundary from the interior during the movement phase are deleted. This is intended for use at 'upstream' or 'inflow' boundaries of the domain, where some features of the incoming flow at the boundary are known.

2. SINK - an 'open' boundary at which no molecules are created. Molecules reaching the boundary from the interior may be deleted (VACUUM subtype) or possibly reflected (PUMP subtype). This is intended for use at 'downstream' or 'outflow' boundaries of the domain.

3. WALL - a solid material boundary at which incident molecules from the interior undergo an arbitrary reflection (or reaction) process. Optionally, molecules may also be created (OUTGAS subtype).

4. SYMMETRY - a boundary at which incident molecules from the interior are specularly reflected. No molecules are created. This is intended for use at symmetry-plane boundaries of the domain.

5. FLUXPLANE - a 'virtual' two-sided boundary plane at which incident molecules are sampled, but undergo no interaction or change in trajectory. No molecules are created. This is intended for use at interior planes or boundaries of the domain at which the user desires detailed information regarding the directed flux.

6. PERIODIC - this consists of two geometrically-equivalent boundaries, where the second is created by a displacement (LINEAR subtype) or rotation (AZIMUTHAL subtype) of the first. Molecules reaching one boundary from the interior undergo a defined displacement to the other boundary, at which they become 'incoming' molecules. No molecules are created.

### 3.5.1 Boundary Condition File Template

```
#### SUPREM Version 1.0 Boundary Condition File ###
user-supplied file description string(s)
ENTITY string
  <COMPONENT  string>
    <SAME_AS [COMPONENT string | ENTITY string | ENTITY string COMPONENT  string]
    <TIME real+>
    TYPE [SOURCE | SINK | WALL | SYMMETRY | FLUX_PLANE | PERIODIC] SUB_TYPE name
     type/subtype-dependent parameters
  <COMPONENT, etc. repeat for remaining components>
<ENTITY, etc. repeat for remaining entities>
#### SUPREM Version 1.0 Boundary Condition File end ###
```

### 3.5.2 Keyword Definition

**ENTITY** *string*

The boundary condition to be defined will be applied to the complete entity named *string*, or to one or more components of the entity referenced on the subsequent line. This name must appear in the geometry definition file (Section 3.3).

**<COMPONENT** *string*>

This optional line identifies the geometrical component named *string* to which the boundary condition will be applied. This name must appear in the geometry definition file (Section 3.3), and must be a component of the referenced entity.

**<SAME_AS** <[COMPONENT *string* | ENTITY *string* | ENTITY *string* COMPONENT *string* ]>

This optional line indicates that the boundary condition will be copied *exactly* as it was defined for the indicated item. The item may be referenced by component name if it is on the same entity, otherwise the original entity name must be included. If this line appears, this is the final entry for the present entity (or component).

**<TIME** *real*>

This line indicates the simulation time at which the following boundary condition is to be activated. This line appears only for boundaries whose properties vary in time. The temporal variation must be representable as a sequence of discrete time-independent boundary condition definitions (ENTITY / COMPONENT / TIME / TYPE ...) ordered by increasing values of the TIME parameter. At a given time, only one member of the sequence may be active. This member is used until the activation TIME of the subsequent member is reached, at which time that boundary becomes active. The presence of any time-dependent boundary requires that the simulation be of type TIME_DEP UNSTEADY (page 36). For this method to be meaningful, the discrete time increments must be larger than the simulation timestep.

**TYPE [SOURCE | SINK | WALL | SYMMETRY | FLUX_PLANE | PERIODIC] SUB_TYPE name**

This identifies the type and subtype of the boundary condition to be applied.

**TYPE SOURCE SUB_TYPE [MACRO | MSIS90 | MANUAL | MOL_LIST | ITERATIVE]**

This defines the boundary as a source of molecules; the molecules are created at a prescribed rate, and with prescribed properties.

**TYPE SOURCE SUB_TYPE MACRO**

This defines the source as a MACROscopic subtype. The user must define several 'bulk' (or mass-averaged, in the case of multi-species) macroscopic

properties of the gas at the boundary, from which the constant equilibrium molecular flux will be calculated analytically. Incoming molecules are sampled from equilibrium distributions which are based on the macroscopic properties. The required inputs are similar to those for the GAS AMBIENT REF_CONDS (page 28):

**[NUMBER_DENSITY** *real*$_+$ | **MASS_DENSITY** *real*$_+$ | **MOLAR_DENSITY** *real*$_+$| **PRESSURE** *real*$_+$]

The user must supply either the number density $n$, mass density $\rho$, or molar density $c$, or the (static) pressure $p$, of the gas at the boundary. These quantities are defined as $\rho = mn$, $p = nk_BT$, $c = \rho/M$, where $m$ is the (mean) molecular mass, $M$ is the (mean) molar molecular weight, $k_B$ is the Boltzmann constant.

**TEMP** *real*$_+$

This is the temperature $T$ at the boundary.

**VEL_X** *real* **VEL_Y** *real* **VEL_Z** *real*

These are the components of the velocity vector $V$ at the boundary.

For a multi-species problem, the user must supply either the mean molecular weight of the gas, $M$, or the representative gas constant $R = k_B/m$ at the boundary, along with the relative species concentrations.

**<[MEAN_MOLEC_WT** *real*$_+$ | **GAS_CONSTANT** *real*$_+$]>

**<SPECIES** *string* [MASS_FRACTION *real*$_+$ | MOLE_FRACTION *real*$_+$ | PARTIAL_PRESSURE *real*$_+$]>

This line defines the relative population of an entering species in a multiple species problem. This line is repeated for each species for which a non-zero entering flux is desired. At least one such line must occur for a multi-species problem. Each species for which a SPECIES entry exists may have optional sampling defined. (It is permissible to define a species flux identically equal to zero, in which case there must be a SAMPLE line for this species, or an input error will be assumed). The quantities for a species $i$ are defined as: mass fraction $\omega_i = \rho_i/\rho$ mole fraction $x_i = c_i/c$, partial pressure $p_i = n_i k_B T$, where $n_i$ and $\rho_i$ are the number density and mass density of the species.

**<SAMPLE** *string* **SPECIES_ID** *int*$_+$ **PROP** name [**AUTO** | **MIN_BIN** *real*$_+$ **MAX_BIN** *real*$_+$ **NUM_BINS** *int*$_+$]>

This optional line activates sampling of the distribution of the named scalar property of the species at the boundary. This line is analogous to the bimolecular collision event SAMPLE line (see Section 3.4.2.2). The

*string* following the SAMPLE keyword is a user-supplied name used only to reference this sample definition (see SAMPLE_BDRY TYPE DETAIL, Section 3.2.2.8). The molecules of this species at the boundary are partitioned into two classes identified by SPECIES_ID - incoming (created, SPECIES_ID is 1) and outgoing molecules (SPECIES_ID is 2). The property to be sampled is identified by its PROPerty name. The sampling range and resolution are determined by the remaining parameters. There can be multiple occurrences of the SAMPLE line per process. Available properties are similar to those for the bimolecular SAMPLE line:

**PROP** [VEL_[X | Y | Z] | ENERGY_TRANS_[X | Y | Z] | ENERGY-modename | ENERGY_TRANS_NORM | ENERGY_TRANS_TAN]

**ENERGY_TRANS_[X** | Y | Z] this is the magnitude of the component of translational energy in the coordinate direction.

**ENERGY_TRANS_NORM** this is the magnitude of the component of translational energy normal to the boundary element.

**ENERGY_TRANS_TAN** this is the magnitude of the component of translational energy tangential to the boundary element.

### TYPE SOURCE SUB_TYPE MSIS90

This defines the source in terms of the MSIS90[] standard atmosphere model. This is intended to be used solely for the case of high speed flight in the earth atmosphere. The user must provide several flight condition parameters from which the constant equilibrium molecular flux will be calculated analytically. Incoming molecules are sampled from equilibrium distributions which are based on macroscopic properties output by the model. This subtype will be documented at a later date.

### TYPE SOURCE SUB_TYPE MANUAL

The manual subtype generalizes the macro subtype to allow the user to prescribe, independently, a spatial variation of properties, a variation in properties between species, and/or a variation (or non-equilibrium) between modes or energy components of each entering species. The format is analogous to the INITIAL_CONDS MANUAL input (page 30). This subtype will be documented at a later date.

### TYPE SOURCE SUB_TYPE MOL_LIST

This defines the source in terms of a list of molecules from which entering molecules are duplicated. The user must define the constant number flux (number of real molecules per unit time per unit area). The molecule list is typically provided by boundary sample output from a prior run. This subtype will be documented at a later date.

### TYPE SOURCE SUB_TYPE ITERATIVE

The ITERATIVE subtype is similar to the MACRO subtype, but it iteratively adjusts the incoming flux to recover a user-supplied net (*incoming − outgoing*) flux at the boundary. This option is intended for boundaries where features of the interior flow may influence incoming flow at the boundary. An example would be a subsonic inflow boundary, where there exists an 'upstream' influence (an upstream-moving characteristic in a continuum flow) which cannot be prescribed using only conditions external to the domain. This boundary condition will be documented at a later date.

### TYPE SINK SUB_TYPE [VACUUM | PUMP]

This defines the boundary as a sink of molecules.

### TYPE SINK SUB_TYPE VACUUM

This defines the sink as a vacuum subtype. All molecules from the interior which intersect (are incident upon) the boundary are deleted. There are no additional inputs required for this type.

### TYPE SINK SUB_TYPE PUMP

This defines the sink as a general 'pumping' surface subtype. Molecules from the interior incident on the boundary are deleted with some finite probability such that the user-defined pumping characteristics are reproduced at the surface. This subtype will be documented at a later date.

### TYPE WALL <SUB_TYPE OUTGAS>

This defines the boundary as a wall. The subtype option OUTGAS is used only when necessary to identify a wall which creates molecules spontaneously (independent of a gas-surface collision event). For each TYPE WALL, the user must define certain basic properties of the wall, such as material type and temperature. The temperature can have a spatial variation. The wall can optionally possess a constant in-plane speed. The user must also define the processes (gas-surface interactions) which an incident molecule may undergo. This definition is analogous to the BIMOLECULAR event definition in the gas file (Section 3.4.2.2).

The template for the TYPE WALL is:

```
TYPE WALL  <SUB_TYPE OUT_GAS>
   MATL string
   TEMP real+
   <XYZ_LIST | NODE_LIST | ELEM_LIST>
     <[XYZ real real real TEMP real+ | NODE int+ TEMP real+ | ELEM int+TEMP real+]>
      <list entries repeat as necessary>
   <VEL [TRANS | ROT]>
     <XYZ_TAIL real real real>
     <XYZ_HEAD real real real>
```

```
  <SPEED real+>
SPECIES string
  string1<()> + MATL → <string3<()> <+ string4<()>> +> MATL
  PROCESS [REFLECTION | REACTION]  MODEL name NUM_PARAMS int+
    <type-dependent parameters>
    <SAMPLE string SPECIES_ID int+ PROP name [AUTO |
              MIN_BIN real+ MAX_BIN real+ NUM_BINS int+]>
    <SAMPLE lines repeat as necessary>
  <chemical equation and PROCESS entries repeat as necessary>
<SPECIES entries repeat as necessary>
```

**MATL** *string*

This defines the wall material as *string*. The *string* or the keyword MATL is used to identify the material in the wall chemical equations in the process definition section. The material name will also be required to reference a materials database in future versions of the code.

**TEMP** $real_+$

This defines the wall surface temperature. It will be applied to the complete referenced boundary. This value can be over-written by an optional spatial variation defined on subsequent lines. The spatial variation is defined in terms of a spatial coordinate list (XYZ_LIST), a node index list (NODE_LIST), or an element index list (ELEM_LIST). The NODE_LIST and ELEM_LIST options are available only for the case of a discretized input geometry. For any list option, each list entry is followed by the temperature to be applied at that location. Element-level entries imply that the temperature is constant over the element, while node-level entries allow for interpolation of properties (regardless of subsequent subdivision of the elements defined by the referenced nodes). The entries are read in the order listed. In the case of overlapping positions or repeated index references, the final occurrence is used.

**<XYZ_LIST>**

This is the header line for the coordinate list option. One or more XYZ and TEMP entries must follow. Currently only a simple linear variation in one coordinate direction is supported. A linear variation in one coordinate direction is assumed if two neighboring XYZ entries have a non-zero value in only one (and the same) coordinate. These coordinates need not lie on the referenced boundary.

**<XYZ** *real real real* TEMP $real_+$**>**

**<NODE_LIST>**

This is the header line for the node index list option. One or more NODE and TEMP entries must follow. This is intended for the case of a relatively general surface temperature distribution defined on the boundary nodes. The temperature at an arbitrary point on

a boundary face element is estimated using a piecewise continuous interpolation from the nodes which compose the element.

<**NODE** *int*$_+$ TEMP *real*$_+$>

This line identifies the index of the node and its temperature.

<**ELEM_LIST**>

This is the header line for the element index list option. One or more ELEM and TEMP entries must follow. This option provides for a (per-element) piecewise constant temperature distribution (no interpolation is performed).

<**ELEM** *int*$_+$TEMP *real*$_+$>

This line identifies the index of the element and its temperature.

<**VEL** [TRANS | ROT]>

This optional line provides for the definition of a restricted class of boundary motion. The motion cannot result in the deformation of the boundary or the domain. It must be describable as a constant 'in-plane' speed at the boundary, and is thus only a simple parameter associated with the molecular interaction at the boundary, rather than with the geometry itself. The motion may be TRANSlational (parameterized by the motion axis [displacement direction] and a speed) or ROTational (parameterized by a rotation axis and a rotational speed). In the case of translational motion, the boundary must be planar and lie on the defined axis. For rotational motion, the boundary must be symmetric about the defined axis. This boundary motion appears simply as a constant vector at each boundary element which is accounted for during calculation of the relative tangential velocity components of the incident or post-interaction molecule.

<**XYZ_TAIL** *real real real* >

This defines the cartesian coordinates of the tail of the vector describing the motion axis.

<**XYZ_TAIL** *real real real* >

This defines the cartesian coordinates of the head of the vector describing the motion axis.

<**SPEED** *real*$_+$>

This defines the magnitude of the speed. For translational motion, the motion is directed from the tail to the head of the motion axis. For rotational motion, the speed is in units of radians per second and the right hand rule is used to determine the direction of motion.

Following the definition of the basic boundary properties, the user must define all processes which can occur for each incident species.

**SPECIES** *string*

This line lists the name of the species for which interaction (collision) equations are to be supplied. The keyword ANY can be used for *string* to denote that the definitions hold for any gas species incident upon the wall (for which explicit entries do not exist). The remainder of the subsection for the given species consists of an arbitrary number of interaction equations, each with an associated PROCESS keyword definition, and an optional SAMPLE keyword.

The interaction equation between the incident species and the material is analogous to that for bimolecular collisions:

*string1*<()> + MATL → <*string3*<()> + *string4*<()>> + MATL

This line defines an abstract interaction process or reaction equation in the form incident species_name(modenames) + MATL → post-interaction species_name(modenames) + MATL. This representation assumes that the actual complex interaction can be approximated as a simpler event involving only the modes listed. An equation which lists the same gas species on each side is a reflection process. All other equations imply a heterogeneous chemical reaction. The event probability may be a constant or depend only on the translational energy and zero or more mode energies which are listed. The pre-collision modes which contribute to the collision energy are listed for the species in the form (modename_i, modename_j, ...), the left-to-right order in which the modes are listed is the order in which the collision energy is incremented (i.e., $e_c = e_t + e_i + e_j + \ldots$).

Post-collision modes which undergo redistribution or reinitialization are denoted by a prime, e.g., (modename1'). The order in which the modes are listed is the order in which the modes are redistributed.

The PROCESS entries and their associated chemical equations define:

1. each potential interaction which may alter one or more modes or states of the incident species (not listed means the interaction cannot and will not occur).
2. which modes contribute to the 'collision energy' or 'available energy' $e_c$.
3. the type of process involved, and the model which will be used to determine the process probability (analogous to a cross section).
4. the method by which the available energy is recalculated and/or redistributed to the (one or more) post-interaction species, to their internal modes, and to translational energy.

Gas-surface interactions are classified into two basic types (PROCESSES):

1. REFLECTION, allowing only the alteration of the translational and/or internal mode energy of the incident molecule. (The alteration is typically a simple reinitialization based on properties of the wall (e.g., temperature), rather than specific details of the collision event (e.g., collision energy) due to deficiencies in available models for this process).

2. REACTION, generalizes (1) to allow the creation, alteration or destruction of species due to the impact or interaction process (heterogeneous chemical reaction). The process probability may be dependent on a collision energy involving an arbitrary combination of the energy modes of the pre-collision molecule, and allows for arbitrary redistribution of the energy into post-interaction or new product species. The number of post-interaction molecules may be zero, one, or two.

Unfortunately, for the case where several PROCESSes have been defined for a given incident species, no consistent, direct analog of the bimolecular null-collision procedure (Section 3.4.2.2) is presently available. In this case, the process which occurs is determined by a combination of the null-collision scheme and the serial-event scheme using the following procedure. A more detailed description of the processes, the models, and their implementation is provided in the Theory manual[].

1. if only REFLECTION processes have been defined, then, with unit probability, each process is carried out serially (independently, and in the order in which the processes are listed). If the chemical equation to which the reflection process definition applies explicitly lists several modes, the process is carried out for each mode in the order listed. In this case, at least one equation must explicitly list translational energy (TRANS). If the chemical equation does not explicitly list any modes (e.g., N2 + MATL → N2' + MATL), the reflection process is carried out for the translational energy (normal component, followed by the two tangential components), followed by each internal mode in the order in which each occurs in the species definition (Section 3.4).

2. if only REACTION processes have been defined, then exactly one reaction will occur. If two or more processes are defined, the selection of the specific reaction process is made using the null-collision scheme.

3. if both REFLECTION and REACTION processes have been defined, either all reflection processes are carried out (as for (1) above), or one reaction process is carried out (as for (2) above). The selection between (1) or (2) is made using the null collision scheme, i.e., the probability of reflection is $p_{(1)} = 1/(1 + p_{(2)})$.

**PROCESS [REFLECTION | REACTION] MODEL name NUM_PARAMS** *int*$_+$

This line defines the process type, model, and model parameters corresponding to the interaction equation. For each PROCESS type, there are one or more pre-defined models, each with a pre-defined number of named parameters. At present, these parameters are restricted to be real-valued scalar constants. Additional models can be added by definition of a model name, the number of parameters required, and the name of each parameter. Each PROCESS entry

can optionally be followed by one or more SAMPLE entries analogous to that for bimolecular sampling (Section 3.4.2.2).

**PROCESS REFLECTION MODEL** [MAXWELL | CLLD ] **NUM_PARAMS** *int*$_+$

A reflection process can utilize the MAXWELL model or the Cercignani-Lampis-Lord (Diffuse) (CLLD) [] model. Each is an empirically-derived analytical model which does not accurately describe the interaction process, but has been found to be useful for engineering analyses involving only reflection processes.

**PROCESS REFLECTION MODEL** MAXWELL **NUM_PARAMS** 2

The basic MAXWELL model assumes that a reflection event is either fully specular or fully diffuse ("diffuse, fully accommodating"). The choice between the two possible post-interaction states is based on an input probability parameter; currently this parameter must be a constant. (Note that, since reaction equations can be made mode-specific, there is no requirement that the same parameter be used for all modes in the incident molecule, and thus this is a generalized form of the Maxwell model). The angular scattering behavior is determined by the parameter used for the translational (kinetic) energy only. A specular reflection results in no change in the magnitude of the energy mode of the incident molecule. A diffuse, fully accommodating reflection is one in which the mode is reinitialized by sampling from an equilibrium distribution characteristic of the local boundary at the impact point; this distribution is described solely by the boundary temperature. This model is thus analogous to the PROCESS INELASTIC MODEL COLLN_NUM for gas collisions (Section 3.4.2.2). The model requires two parameters: the probability that the reflection is diffuse, and a reference temperature at which this probability was evaluated.

**PROB_DIFFUSE** *real*

This is the probability that the reflection is diffuse, $0 \leq P_d \leq 1$. A value of zero means the reflection is specular for the mode in question.

**TEMP_REF** *real*$_+$

This is a reference temperature $T_{\text{ref}}$ at which $P_d$ was evaluated. This is used solely to evaluate the applicability of the model as applied to this boundary.

**[COMMENT** *string*]

This is an optional comment.

**PROCESS REFLECTION MODEL CLLD NUM_PARAMS 2**

The CLLD (Cercignani-Lampis-Lord, Diffuse) model provides for diffuse angular scattering but incomplete (relative to the surface equilibrium value) energy accommodation. The model requires two parameters: the energy "accommodation coefficient" or probability of accommodation, and a reference temperature at which this coefficient was evaluated.

**ACCOM** *real*

This is the accommodation coefficient $a_c$, $0 \leq a_c \leq 1$ for the mode(s) involved. This is a measure of the degree to which the reflected mode energy is equilibrated toward the wall value. The diffuse, fully accommodating model (equivalent to Maxwell, is recovered for $a_c = 1$.

**TEMP_REF** *real*$_+$

This is a reference temperature $T_{\text{ref}}$ at which $a_c$ was evaluated. This is used solely to evaluate the applicability of the model as applied to this boundary.

**[COMMENT** *string*]

This is an optional comment.

**PROCESS REACTION MODEL** [ADSORB | CATALYTIC] **NUM_PARAMS** *int*$_+$

At present only a restricted class of relatively simple reaction processes are allowed. Others will be added in the future.

**PROCESS REACTION MODEL** ADSORB **NUM_PARAMS** 2

This model allows an incident molecule to adsorb (stick) on the surface. The adsorption probability is supplied as an input parameter; currently this parameter must be a constant. Molecules which adsorb are permanently deleted from the simulation (there is no reverse, or desorption event). This model is intended to allow for qualitative simulation of condensing surfaces such as a cryopanel. A REFLECTION process must be defined for the incident species.

**PROB_ADSORB** *real*$_+$

This is the probability that the molecule is adsorbed (i.e., the sticking coefficient), $0 < P_a \leq 1$.

**TEMP_REF** *real*$_+$

This is a reference temperature $T_{\text{ref}}$ at which $P_a$ was evaluated. This is used solely to evaluate the applicability of the model as applied to this boundary.

**[COMMENT** *string*]

This is an optional comment.

**PROCESS REACTION MODEL** CATALYTIC **NUM_PARAMS** 2

This model allows an incident molecule to undergo an instantaneous catalytic reaction, resulting in the creation of a single molecule of a different species which leaves the surface. The catalytic probability is supplied as an input parameter; currently this parameter must be a constant. The new molecule is initialized assuming full accommodation to the surface. A REFLECTION process must be defined for the incident species.

**PROB_CATALYTIC** $real_+$

This is the probability that the molecule undergoes catalytic reaction, $0 < P_c \leq 1$.

**TEMP_REF** $real_+$

This is a reference temperature $T_{\text{ref}}$ at which $P_c$ was evaluated. This is used solely to evaluate the applicability of the model as applied to this boundary.

**[COMMENT** *string*]

This is an optional comment.

**<SAMPLE** *string* **SPECIES_ID** $int_+$ **PROP** name [**AUTO** | **MIN_BIN** $real_+$ **MAX_BIN** $real_+$ **NUM_BINS** $int_+$]>

This optional line activates sampling of the distribution of the named scalar property of a particular species involved in the process. This line is analogous to the bimolecular collision event SAMPLE line (see Section 3.4.2.2). The *string* following the SAMPLE keyword is a user-supplied name used only to reference this sample definition. The species to be sampled is identified by the index of occurrence of the species in the collision equation (SPECIES_ID is 1 for incident (pre-interaction) species, or 2 [or greater, for certain reaction processes] for the post-interaction species). The property to be sampled is identified by its PROPerty name. The sampling range and resolution are determined by the remaining parameters. There can be multiple occurrences of the SAMPLE line per process. Available properties are similar to those for the bimolecular SAMPLE line:

**PROP** [VEL_[X | Y | Z] | ENERGY_TRANS_[X | Y | Z] | ENERGY-modename | ENERGY_COLLN | ENERGY_TRANS_NORM | ENERGY_TRANS_TAN | ANGLE_NORM]

**ENERGY_TRANS_[X** | Y | Z] this is the magnitude of the component of translational energy in the given coordinate direction.

**ENERGY_TRANS_NORM** this is the magnitude of the component of translational energy normal to the boundary element.

**ENERGY_TRANS_TAN** this is the magnitude of the component of translational energy tangential to the boundary element.

**ANGLE_NORM** this is the angle of the velocity vector measured with respect to the boundary element normal.

**TYPE** WALL SUB_TYPE OUTGAS

For a wall which can outgas, or spontaneously create molecules at the surface which then enter the domain, additional quantities must be defined. This subsection describes these parameters. They are analogous to those for TYPE SOURCE SUB_TYPE MACRO (page 62). This subtype option will be documented at a later date.

**TYPE SYMMETRY**

This identifies the symmetry type, for which there are no subtypes available. There are no additional inputs required for this type. This can only be applied to boundaries which are planar.

**TYPE FLUX_PLANE**

This defines the boundary as a flux plane, for which there are no subtypes. This can only be applied to boundaries which are planar. The user may define the properties to be sampled for molecules which are incident upon either side of the plane.

**TYPE PERIODIC SUB_TYPE [LINEAR | AZIMUTHAL]**

This defines the boundary as a periodic type. This option will be documented at a later date.

Intentionally left blank.

# Chapter 4

# Code Operation

Figure 4.1 shows a flowchart of the operation of the SUPREM package.

## 4.1 Pre-processing with Pre-SUPREM

The pre-processor reads the basic input files defined in Chapter 3 and converts them to a form suitable for input to SUPREM. This code is typically run interactively. The processing includes extensive error checking and attempts to differentiate between invalid input data (e.g., incorrect keywords or related parameters) and acceptable, but potentially inaccurate, input data. Successful completion of pre-processing results in the creation of (1) the three restart files required by SUPREM, (2) a variety of informational output summarizing the input data and the resulting initial simulation parameters, and (3) additional output providing suggested input values for the SUPREM code.

The pre-processor can also be used to refine an existing simulation. In this case one or more of the restart files must be available, along with the basic input files.

## 4.2 SUPREM

## 4.3 Post-processing with Post-SUPREM

Figure 4.1: Flowchart for operation of the SUPREM package. (a) pre-processing with Pre-SUPREM, (b) running SUPREM, (c) post-processing with Post-SUPREM, (d) restarting an existing SUPREM run, (e) refining an existing SUPREM run via Pre-SUPREM.

Intentionally left blank.

# Chapter 5

# Demonstration Problems

5.1 Reentry Vehicle Flowfield

5.2 Microvalve Flow

no test for ch. 5?

# Chapter 6

# Frequently Asked Questions, Tips and Troubleshooting

No text for ch. 6?

# Chapter 7

# Utilities

A variety of utilities are provided for use with the package, in the form of shell scripts or fortran90 source code. These utilities can be grouped into several classes:

1. datafile transformation (e.g., conversion of a CAD geometry file to SUPREM geometry format).
2. data manipulation (e.g., conversion of dimensional units used in various input files).

Intentionally left blank.

# Chapter 8

# Acknowledgement

no acknowledgement?

# Chapter 9

# Bibliography

no bibli ?

# Appendix A

# Sample Input Files

A.1 Execution Control (.in) Files

A.2 Geometry (.geo) Files

A.3 Gas (.gas) Files

A.4 Boundary Condition (.bc) Files

# Appendix B

# Sample Output Files

B.1 Pre-Processor Output

B.2 SUPREM Output

B.3 Post-Processor Output

# Appendix C

# Error Messages

Error messages are grouped into three classes: fatal, severe, and warning. The messages are intended to be self-explanatory.

A fourth class, informational, is used to provide informative messages only.